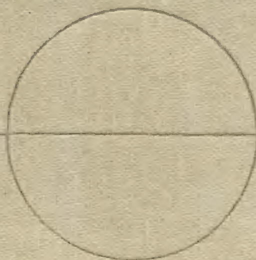
VISIT…

LANZAROTE Caliente.COM

512.8
Б
22.18
Б-41

И. В. БЕЙКО
Б. Н. БУБЛИК
П. Н. ЗИНЬКО

МЕТОДЫ И АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ



И. В. БЕЙКО,
Б. Н. БУБЛИК,
П. Н. ЗИНЬКО

МЕТОДЫ И АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ ОПТИМИЗАЦИИ

КИЕВ
ГОЛОВНОЕ ИЗДАТЕЛЬСТВО
ИЗДАТЕЛЬСКОГО ОБЪЕДИНЕНИЯ
«ВИЩА ШКОЛА»
1983

32.965.9
Б77

УДК 519.9

Методы и алгоритмы решения задач оптимизации. Бейко И. В., Бублик Б. Н., Зинько П. Н.— К.: Вища школа. Головное изд-во, 1983.— 512 с.

В справочном пособии изложены современные методы и алгоритмы для решения задач оптимизации, возникающих во многих областях науки и техники, в сфере управления экономическими, социальными, техническими и другими процессами. Рассмотрены линейные и нелинейные, детерминированные и стохастические, гладкие и негладкие, минимаксные и другие задачи оптимизации. Все методы оптимизации представлены в виде детально разработанных алгоритмов. Для облегчения поиска необходимого алгоритма и его практического использования приводится независимое описание каждого метода, включающее постановку задачи оптимизации, ограничительные предположения, описание конкретных алгоритмов и соответствующих теорем сходимости, а также необходимые библиографические указания.

Книга рассчитана на работников научно-исследовательских учреждений и вычислительных центров, занимающихся вопросами разработки и применения методов оптимизации, а также на студентов, специализирующихся по прикладной математике и другим специальностям, связанным с использованием ЭВМ.

Библиогр.: 575 назв.

Рецензенты: академик АН УССР *В. С. Михалевич* и канд. физ. мат. наук *М. Я. Бартиш* (Львовский государственный университет)

Редакция литературы по кибернетике, электронике и энергетике

Зав. редакцией *М. С. Хойнацкий*

Б $\frac{1502010000-222}{М211(04)-83}$ 95—83

© Издательское объединение «Вища школа», 1983

ОГЛАВЛЕНИЕ

Обозначения и символы 9

Предисловие 11

Введение. **Элементы теории оптимизации и управления** 13

0.1. Математические методы управления и задачи оптимизации 13

1. О математических методах поиска оптимальных решений (13). 2. Основные задачи оптимизации (14). 3. О задачах многоцелевого управления (19).

0.2. О некоторых методах решения задач оптимизации 20

1. Метод полного перебора (21). 2. О приближенных методах и оптимальных алгоритмах (21).

0.3. Методы отсечений 24

1. Минимизация квазивыпуклых функций (24). 2. Метод минорант. Минимизация липшицевых функций (25). 3. Минимизация выпуклых функций. Метод эллипсоидов (25).

0.4. О локализации решений с помощью необходимых и достаточных условий оптимальности. Дополнительная терминология 27

1. Необходимые условия оптимальности для дифференцируемых функций (27). 2. Необходимые и достаточные условия оптимальности для задач выпуклого программирования. Теорема Куна — Таккера (28). 3. Необходимые условия первого и второго порядка для задач нелинейного программирования (30). 4. Условия оптимальности для задач минимизации негладких функций (31).

0.5. Методы последовательных приближений 32

1. О способах выбора шаговых множителей в задачах безусловной оптимизации (33). 2. О методах второго порядка (36). 3. Методы сопряженных градиентов (38). 4. О методах штрафов в задачах с ограничениями (39). 5. О методах последовательных приближений для задач условной оптимизации и минимизация негладких функций (40).

Часть I

МЕТОДЫ ОДНОМЕРНОЙ И БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

Глава 1. **Методы одномерной оптимизации** 46

1.1. Методы Фибоначчи 46

1. Основной алгоритм (46). 2. Модификация метода Фибоначчи (47).

1.2. Метод золотого сечения 48

1.3. Оптимальный метод поиска экстремума унимодальных функций, удовлетворяющих условию Липшица 49

1.4. Оптимальный метод поиска экстремума выпуклых функций 52

1.5. Методы типа Ньютона 54

1 Метод Ньютона (55). 2. Метод секущей (55).

1.6. Методы касательных 56

1. Случай дифференцируемой функции (56). 2. Случай недифференцируемой функции (57).

1.7. Метод квадратичной аппроксимации 58

1.8. Метод отыскания абсолютного минимума функций, удовлетворяющих условию Липшица 59

1.9. Метод кусочно-кубической аппроксимации 61

1.10. Методы глобального поиска 63

1. Алгоритм глобального поиска (63). 2. Рандомизированный алгоритм глобального поиска (65).

1.11. Методы поиска интервала наибольших значений многоэкстремальных функций 66

1.12. Методы поиска глобального минимума, использующие стохастические автоматы 68

1. Алгоритм, использующий модель Буша — Мостеллера (69). 2. Алгоритм, использующий усредненные значения функции (70).

1.13. Адаптивные методы 71
1. Алгоритмы Кифера — Вольфовица (71). 2. Простой перебор (72).

Глава 2. **Методы оптимизации дифференцируемых функций** 73

2.1. Градиентные методы 73
1. Метод наискорейшего спуска (73). 2. Модифицированный метод наискорейшего спуска (74). 3. Основной вариант градиентного метода (76). 4. Градиентный метод с постоянным шаговым множителем (77). 5. Вариант градиентного метода с матрицей ускорения сходимости (78). 6. Модифицированный градиентный метод, не требующий вычисления производных (80).

2.2. Методы типа Ньютона 82
1. Метод Ньютона — Канторовича (82). 2. Обобщенный метод Ньютона — Канторовича (82). 3. Модификации обобщенного метода Ньютона — Канторовича (84). 4. Модифицированный обобщенный метод Ньютона — Канторовича, не требующий вычисления матрицы вторых производных (85)

2.3. Методы двойственных направлений 87
1. Основной вариант (87). 2. Модифицированный метод двойственных направлений, не требующий вычисления производных (89).

2.4. Методы сопряженных градиентов 92
1. Общая схема алгоритмов сопряженных градиентов (92). 2. Метод сопряженных градиентов с восстановлением (93). 3 Реализуемые модификации алгоритмов сопряженных градиентов (94) 4. Метод сопряженных градиентов для минимизации квадратичных функций (97). 5. Реализуемая модификация алгоритма с переменной метрикой (98).

2.5. Методы сопряженных направлений 99
1. Метод сопряженных направлений с восстановлением матрицы (99). 2. Метод сопряженных направлений без восстановления матрицы (100). 3. Минимизация квадратичных функций с помощью метода сопряженных направлений (102). 4. Модифицированный метод сопряженных направлений, не требующий вычисления производных (105).

2.6. Методы псевдообратных операторов 106
1 Основной алгоритм (107). 2. Устойчивое псевдообращение плохо обусловленных матриц (110).

2.7. Методы минимизации вдоль собственных векторов матрицы, близкой к матрице Гессе 112
1. Основной вариант (113). 2. Ускоренный вариант (115).

2.8. Итеративные стохастические методы, использующие аналог функции Ляпунова 116
1. Основной алгоритм (117). 2. Сходимость алгоритма в среднем (117). 3. Сходимость алгоритма почти наверное (120). 4. Модификации алгоритма (121)

2.9. Стохастические квазиградиентные методы 121
1. Общий стохастический квазиградиентный метод для детерминированных задач (121). 2. Общий стохастический квазиградиентный метод для стохастических задач (123). 3. Стохастический квазиградиентный метод с процедурой прерывания (124). 4. Стохастический квазиградиентный метод с постоянным шаговым множителем (125). 5. Стохастический градиентный метод минимизации сложных функций регрессии (126).

2.10. Метод локальных вариаций 128
2.11. Методы самонастраивающихся программ 129
2.12. Общий метод спуска 132
2.13 Методы случайного поиска 133
1. Алгоритм случайного поиска в выпуклых задачах минимизации (133). 2. Адаптивный алгоритм случайного поиска (135).

Глава 3. **Методы оптимизации недифференцируемых функций и методы отыскания седловых точек** 136

3.1. Методы обобщенного градиентного спуска 136
1. Алгоритм с постоянным шаговым множителем (136). 2. Основной алгоритм (137). 3. Модификация основного алгоритма (138). 4. Первый алгоритм со специальным выбором шага (139). 5 Второй алгоритм со специальным выбором шага (140). 6. Алгоритм, использующий априорное знание минимума функции (141). 7. Помехоустойчивый алгоритм (142). 8. Многошаговый метод обобщенного градиентного спуска (143). 9. ε-субградиентный метод (144).

3.2. Методы градиентного типа с растяжением пространства 146
3.3. Методы градиентного типа с растяжением пространства в направлении разности двух последовательных почти градиентов ($r(\alpha)$-алгоритм) 149
3.4. Методы локального случайного поиска 151

1. Алгоритм локального случайного поиска с парной пробой (151). 2. Алгоритм локального случайного поиска с возвратом при неудачном шаге (152). 3. Алгоритм локального случайного поиска с линейной экстраполяцией (152). 4. Алгоритм случайного поиска по наилучшей пробе с накоплением (153). 5 Алгоритм статистического градиента (153)

3.5. Псевдоградиентные методы адаптации и обучения 154
3.6. Квазиградиентные методы 156

1. Квазиградиентный метод минимизации слабовыпуклой вниз функции (157). 2. Стохастический квазиградиентный метод минимизации слабовыпуклой функции (158)

3.7. ε-квазиградиентные методы 159

1. ε-квазиградиентный метод минимизации выпуклых функций (159). 2. ε-квазиградиентный метод минимизации слабовыпуклых функций (160).

3.8. Методы обобщенных почти градиентов для минимизации функций, удовлетворяющих локальному условию Липшица 161

1. Детерминированный случай (161). 2. Стохастический случай (163).

3.9. Метод усреднения направлений спуска для минимизации функций, удовлетворяющих условию Липшица 164
3.10 Конечно-разностный метод минимизации разрывных функций 165
3.11. Метод линеаризации решения дискретных минимаксных задач 167
3.12. Методы последовательных приближений для решения дискретных минимаксных задач 169

1. Первый метод последовательных приближений (169). 2. Модификация первого метода последовательных приближений (172). 3. Второй метод последовательных приближений (174). 4. Модификация второго метода последовательных приближений (175). 5. Третий метод последовательных приближений, использующий D-функцию (177)

3.13. Сеточный метод последовательных приближений решения непрерывных минимаксных задач 178
3.14. Методы Эрроу — Гурвица решения непрерывных минимаксных задач 180

1. Детерминированный метод Эрроу — Гурвица (180). 2. Стохастический метод Эрроу — Гурвица (181)

3.15. Методы экстремального базиса для решения непрерывных минимаксных задач 182

1. Принципиальный алгоритм (182). 2. ε-метод (184). 3. ρ-метод (185). 4. Комбинированный ε, ρ-метод (185).

3.16. Обобщенный градиентный метод отыскания седловых точек 187
3.17. Квазиградиентные методы решения дискретных минимаксных задач стохастического программирования 188

1. Минимизация функции $E_\omega \max\limits_{i\in J} \varphi_i(x, \omega)$ (188). 2. Минимизация функции $\max\limits_{i\in J} E_\omega \varphi_i(x, \omega)$ (190).

Часть II

МЕТОДЫ УСЛОВНОЙ ОПТИМИЗАЦИИ 192

Глава 4. **Методы решения задач линейного программирования** 192
4.1. Симплекс-метод и его варианты 193

1. Симплекс-метод решения невырожденной канонической задачи линейного программирования (194). 2. Методы отыскания исходного базиса (196). 3. Симплекс-метод решения вырожденной канонической задачи линейного программирования (199). 4. Модифицированный симплекс-метод (201).

4.2. Двойственный симплекс-метод 203
1. Основной алгоритм (203). 2. Методы отыскания исходного опорного решения сопряженной задачи (207). 3. Правило выбора вектора a^s, вводимого в базис, гарантирующее от зацикливания в вырожденном случае (210).
4.3. Методы последовательного сокращения невязок 211
1. Метод последовательного сокращения невязок (211). 2. Метод двусторонних оценок (214).
4.4. Обобщенный симплекс-метод 216
4.5. Методы блочного программирования 219
1. Метод разложения Данцига — Вулфа (219). 2. Метод разложения решения задач линейного программирования с блочно-диагональной матрицей (223). 3. Метод, использующий обобщенный градиентный спуск (227).
4.6. Модификация симплекс-метода для решения задачи линейного программирования с двусторонними ограничениями 229
1. Основной алгоритм (230). 2. Правило выбора индекса r (определяющего вектор a^{i_r}, выводимый из базиса) для предотвращения зацикливания (233).
4.7. Модификация симплекс-метода для решения общей задачи линейного программирования 235
4.8. Итеративные методы 240
1. Итеративный метод Петшиковского (240). 2. Итеративный метод, использующий модифицированную функцию Лагранжа (242). 3. Итеративный метод Федоренко (243). 4. Алгоритм «Заяц» решения прямой и двойственной задач линейного программирования (247). 5. Итерационный метод, использующий модифицированную функцию Лагранжа для решения двойственной пары задач линейного программирования (250).
4.9. Методы параметрического программирования 253
1. Случай наличия параметра в целевой функции (253). 2. Случай наличия параметра в правых частях ограничений (257).
4.10. Опорные методы 261
1. Прямой опорный метод (262). 2. Метод обратной матрицы (265).
Глава 5. **Методы решения задач нелинейного и стохастического программирования** 269
5.1. Методы проекции градиента 269
1. Общий алгоритм (269). 2. Метод проекции градиента для минимизации функций при линейных ограничениях (270). 3. Гибридный метод проекции градиента для минимизации функций при нелинейных ограничениях (274) 4. Метод проекции градиента при наличии возмущений (277).
5.2. Общий метод штрафных функций 278
5.3. Методы внешних штрафных функций 281
1. Задачи с ограничениями в виде неравенств (281). 2. Задачи с ограничениями в виде равенств (283). 3. Модифицированные методы с процедурой прерывания (284). 4. Метод внешних штрафных функций для минимизации недифференцируемых функций (287).
5.4. Методы внутренних штрафных функций 289
1. Общая схема (289). 2. Реализуемая схема с процедурой прерывания (290). 3. Алгоритмы внутренней точки с применением Q-функций (292).
5.5. Комбинированные методы штрафных функций 293
5.6. Стохастический метод штрафов 295
5.7. Методы возможных направлений 297
1. Методы возможных направлений решения задач минимизации с ограничениями типа неравенств (297). 2. Методы возможных направлений решения задач минимизации с ограничениями смешанного типа (301) 3. Метод возможных направлений с квадратичным поиском (305). 4. Модифицированный метод возможных направлений Зойтендейка (307) 5. Аналог метода возможных направлений в задачах минимизации почти дифференцируемых функций (309). 6. Стохастический аналог метода возможных направлений (311). 7. Методы возможных направлений для отыскания точек локальных минимумов невыпуклой функции на невыпуклом множестве (313).
5.8. Методы центров 315
1. Основной вариант (316). 2. Модифицированный метод центров (316). 3. Реализация модифицированного метода центров с использованием метода золотого сечения (318). 4. Реализация модифицированного метода центров с использованием функций прерывания (319).

5.9. Методы чебышевских центров 320
1. Метод чебышевских центров (320). 2. Модифицированный метод чебышевских центров (321).
5.10. Методы типа Ньютона 322
1. Метод типа Ньютона с регулировкой шага (322). 2. Метод типа Ньютона при наличии возмущений (323). 3. Квазиньютоновские методы (325).
5.11. Методы линеаризации 327
1. Ограничения типа неравенств (328). 2. Ограничения типа равенств (330) 3. Ограничения смешанного типа (332). 4. Метод линеаризации, практически реализуемый на ЭВМ (333). 5. Аналог метода линеаризации в детерминированных задачах минимизации почти дифференцируемых функций (335). 6. Аналог метода линеаризации в стохастических задачах минимизации почти дифференцируемых функций (336). 7. Стохастический метод линеаризации (337).
5.12. Методы линеаризации в предельных экстремальных задачах 339
1. Детерминированный случай (339). 2. Стохастический случай (341).
5.13. Методы отсечения 343
1. Линейный случай (343). Общий случай (345). 3. Метод отсечения с растяжением пространства для решения задач выпуклого программирования (346).
5.14. Методы, использующие функцию Лагранжа 350
1. Градиентный метод для задач с ограничениями типа неравенств (350). 2. Градиентный метод для задач с ограничениями типа равенств (351). 3. Метод квадратичной аппроксимации для задач с ограничениями типа равенств (353). 4. Двойственный метод для задач с ограничениями типа равенств (353). 5. Метод Ньютона для задач с ограничениями типа равенств (354).
5.15. Методы, использующие модифицированные функции Лагранжа . . 355
1. Градиентный метод (355). 2. Метод, использующий штрафные функции экспоненциального типа (357).
5.16. Методы нагруженного функционала 360
1. Общий случай (360). 2. Выпуклый случай (361). 3. Приближенная схема (363).
5.17. Методы штрафных оценок 365
1. Детерминированный случай (365). 2. Стохастический случай (367). 3. Метод штрафных оценок для задач выпуклого программирования (368).
5.18. Методы проектирования обобщенного градиента 371
1. Основной алгоритм (371). 2. Многошаговый метод обобщенного градиентного спуска (372). 3. Методы проектирования обобщенного градиента для решения предельных экстремальных задач (373).
5.19. Методы условного градиента 375
1. Реализуемый метод условного градиента (375). 2. Алгоритм Франка — Вулфа (376). 3. Ускоренный алгоритм Франка — Вулфа (377).
5.20. Методы сопряженных градиентов 378
1. Метод сопряженных градиентов (378). 2. Стохастический аналог метода сопряженных градиентов (380).
5.21. Методы управляющих последовательностей 381
1. Общий метод минимизации при наличии ограничений (382). 2. Метод управляющих последовательностей (383).
5.22. Методы внутренней аппроксимации 387
5.23. Методы покоординатного спуска 388
1. Детерминированный покоординатный спуск (388). 2. Случайный покоординатный спуск (390).
5.24. Релаксационные методы для общих задач нелинейного программирования 392
1. Непрерывный вариант (392). 2. Дискретный вариант (393).
5.25. Методы фейеровских приближений решения задач выпуклого программирования с негладкими ограничениями 395
1. Общий случай (395). 2. Случай кусочно линейных ограничений (396).
5.26. Двойственные методы 398
1. Приближенный двойственный метод решения задач выпуклого программирования (398). 2. Двойственный алгоритм переменной метрики (401).

5.27. Глобально сходящийся метод 403
5.28. Стохастические квазиградиентные методы 405
1. Метод проектирования стохастических квазиградиентов (407). 2. Стохастический метод сокращения невязок в детерминированных задачах (410). 3. Метод сокращения невязок в задачах стохастического программирования (412). 4. Гибридный стохастический метод (414)
5.29. Комбинированный метод стохастических градиентов и штрафных функций 416
5.30. Методы усреднения направлений спуска 417
1. Детерминированная задача (417). 2. Стохастическая задача (420).
5.31. Прямой метод решения задач стохастического программирования . . 421
5.32. Метод случайного поиска в выпуклых задачах минимизации 423
5.33. Методы решения задач оптимизации с бесконечным числом ограничений 424
1. Общий алгоритм (425). 2. Ослабленный алгоритм (426).
5.34. Методы решения задач квадратичного программирования 427
1. Метод сопряженных градиентов для минимизации квадратичной функции на подпространстве (427). 2. Метод сопряженных градиентов для общей задачи квадратичного программирования (428). 3. Метод сопряженных градиентов для задачи квадратичного программирования с простыми ограничениями (431). 4. Модификация метода сопряженных направлений для задач квадратичного программирования большой размерности (432). 5. Устойчивый алгоритм решения задач квадратичного программирования (435).

Глава 6. **Специальные методы решения минимаксных задач и методы отыскания седловых точек** 438
6.1. Методы последовательных приближений решения дискретных минимаксных задач 438
1. Минимаксная задача с ограничениями простой структуры (438). 2. Первый метод последовательных приближений решения минимаксной задачи с ограничениями типа неравенств (440). 3. Второй метод последовательных приближений решения минимаксной задачи с ограничениями типа неравенств (442). 4. Модификация второго метода последовательных приближений (443).
6.2. Обобщенный беспараметрический метод внешней точки решения дискретных минимаксных задач 444
1. Основной алгоритм (444). 2. Ускоренный вариант алгоритма (445).
6.3. Метод второго порядка решения задач дискретного минимакса . . . 446
6.4. Сеточный метод последовательных приближений решения непрерывных минимаксных задач 449
6.5. Метод штрафа в задаче поиска максимина 450
6.6. Методы стохастического квазиградиента в задаче поиска максимина 452
1. Основной алгоритм (453). 2. Следящий алгоритм (455). 3. Выпуклый случай (457).
6.7. Метод невязок в задаче поиска максимина 458
6.8. Квазиградиентные методы решения непрерывных минимаксных задач стохастического программирования 461
1. Стохастический квазиградиентный метод (461). 2. Модифицированный стохастический квазиградиентный метод (462).
6.9. Градиентные методы отыскания седловых точек 464
1. Основной алгоритм (465). 2. Градиентный метод отыскания седловых точек с постоянным шаговым множителем (465). 3. Обобщенный градиентный метод (467).
6.10. Метод экспоненциальных штрафов отыскания седловых точек 468
6.11. Задачи оптимального управления и оценивания. Методы развязывающих операторов 471
Список литературы 482
Предметный указатель 506

ОБОЗНАЧЕНИЯ И СИМВОЛЫ

R — множество действительных чисел

α, β, γ, δ, ε, λ, μ, ν, ρ, σ, τ — действительные числа

i, j, k, l, m, n, s — целые числа

R^n — n-мерное евклидово пространство, т. е. множество векторов $x = (x_1, x_2, \ldots, x_n)$, все компоненты которых $x_1, x_2, \ldots, x_n$ являются действительными числами

$a, b, c, d, e, f, g, h, x, y, z, \ldots$ и $a^i, b^l, a^k, x^m, \ldots$ — векторы, т. е. $a \in R^n$, $x^m \in R^k$, ...

A, B, C, H — матрицы

a_{ij}, b_{ij}, c_{ij} и h_{ij} — элементы i-й строки и j-го столбца соответственно матриц A, B, C и H

$\mathcal{I}$, $\mathcal{K}$, L — множество индексов (целых чисел)

(x, y) — скалярное произведение векторов x и y, т. е. $(x, y) = \sum_i x_i y_i$

$\|x\|$ — евклидова норма вектора x, т. е. $\|x\| = (x, x)^{1/2}$

f_i — скалярная функция

$f_i : X \to Y$ — функция, определенная на множестве X и принимающая значения из множества Y

$f : X \to R^n$ — n-мерная вектор-функция $f = (f_1, f_2, \ldots, f_n)$; во введении f — скалярная функция

A^T — матрица, транспонированная к матрице A

A^{-1} — матрица, обратная к матрице A

$\nabla f_i(x)$ — градиент функции f_i в точке x, т. е.

$$\nabla f_i(x) = \left(\frac{\partial f_i(x)}{\partial x_1}, \ldots, \frac{\partial f_i(x)}{\partial x_n} \right)$$

$\nabla f(x)$ — матрица Якоби для вектор-функции f в точке x, т. е. элемент i-й строки и j-го столбца данной матрицы равен частной производной $\partial f_i(x)/\partial x_j$

$\nabla^2_{xx} f_i(x)$ — матрица Гессе для функции f_i в точке x, т. е. элемент k-й строки и j-го столбца данной матрицы равен $\partial^2 f_i(x)/\partial x_k \partial x_j$

$\hat{\nabla} f_i(x)$ — обобщенный градиент функции f_i в точке x

$\tilde{\nabla} f_i(x)$ — квазиградиент функции f_i в точке x

$G \triangleq Q$ — G равно Q по определению (G эквивалентно Q)

$G \supset Q$ — G содержит Q

$G \subset Q$ — G содержится в Q

$G \cup Q$ — объединение G и Q

$G \cap Q$ — пересечение G и Q

$G \times Q$ — декартово произведение G и Q

$\{x \mid G\}$ — множество всех x, для которых верно утверждение G

$x \neq y$ — x не равно y

$x \in Q$ — x принадлежит Q

$x \notin Q$ — x не принадлежит Q

$K \not\ni 0$ — множество K не содержит индекс 0

$\varnothing$ — пустое множество
I — единичная матрица
(α, β) — открытый интервал $\{x \mid \alpha < x < \beta\}$
$[\alpha, \beta]$ — замкнутый интервал $\{x \mid \alpha \leqslant x \leqslant \beta\}$
$\max\limits_{i \in \mathcal{I}}$ — максимум по всем i, принадлежащим множеству $\mathcal{I}$
$x \leqslant y$ — $x \leqslant y$ для $x \in \boldsymbol{R}^n$, $y \in \boldsymbol{R}^n$, если $x_i \leqslant y_i$ для $i = 1, \ldots, n$
$\boldsymbol{R}^n_+$ — множество $\{x \mid x \geqslant 0, \ x \in \boldsymbol{R}^n\}$
$\operatorname{diam} Y$ — диаметр множества Y
$\operatorname{int} X$ — внутренность множества X
$\operatorname{tr} A$ — след матрицы A
$\operatorname{rang} A$ — ранг матрицы A
$\operatorname{Ent}(t)$ — целая часть числа t
$R(A)$ — образ матрицы A
$[1:n]$ — множество целых чисел от 1 до n включительно
$\operatorname{co} X$ — выпуклая оболочка множества X
$\det A$ — определитель матрицы A
$O(\alpha)$ — величина порядка α
$o(\alpha)$ — величина бесконечно малая по сравнению с α
$(b \,\vdots\, A)$ — матрица, образованная столбцом b и столбцами матрицы A
$\overset{\text{п.н}}{=}$ — равно почти наверное
$[t]_+$ — $[t]_+ = \max\{0, t\}$
$(\beta_{ij})_{i=1,\ldots,m}^{j=1,\ldots,n}$ — матрица размера $m \times n$, (i, j)-м элементом которой является β_{ij}
$(f_j, j \in \mathcal{I})$ — вектор, компонентами которого являются числа f_j, $j \in \mathcal{I}$
$\{x, y, \ldots, z\}$ — множество, состоящее из элементов $x, y, \ldots, z$
$\max\{t_1, t_2, \ldots, t_n\}$ — максимальный элемент из множества $\{t_1, t_2, \ldots, t_n\}$
$\boldsymbol{P}$ {событие S} — вероятность события S
$\forall x \in X$ — для всех элементов x из множества X
$\exists x \in X$ — существует такой элемент x в множестве X
$\arg\min\limits_{x \in X} f_i(x)$ — тот элемент x^* из множества X, который доставляет наименьшее значение $f_i(x^*)$ функции f_i на множестве X, т. е.

$$\forall x \in X \quad f_i(\arg\min_{x \in X} f_i(x)) \leqslant f_i(x)$$

$\operatorname{Arg}\min\limits_{x \in X} f_i(x)$ — множество всех элементов $\arg\min\limits_{x \in X} f_i(x)$
$\arg\max\limits_{x \in X} f_i(x)$ — элемент x^{**} из множества X, который доставляет наибольшее значение $f_i(x^{**})$ функции f_i на множестве X, т. е.

$$\forall x \in X \quad f_i(\arg\max_{x \in X} f_i(x)) \geqslant f_i(x)$$

X^* — множество решений задачи оптимизации
$\boldsymbol{E}\xi$ — математическое ожидание случайной величнны ξ
$\boldsymbol{E}(\xi/x)$ — условное математическое ожидание при данном x
$\boldsymbol{D}\xi$ — дисперсия случайной величины ξ
Ω — пространство элементарных событий ω

ПРЕДИСЛОВИЕ

Одним из наиболее интенсивно используемых и наиболее важных инструментариев повышения эффективности управления и оптимизации сложных систем являются в настоящее время математические методы оптимизации.

Современные методы оптимизации часто оказываются недоступными для многих потенциальных потребителей из-за высокого математического уровня соответствующих публикаций. Поэтому при написании данного пособия авторы стремились довести до реализуемых алгоритмов многие из современных методов оптимизации и сделать пособие удобным для студентов и широких кругов специалистов, использующих методы оптимизации в различных областях науки и техники. Этими целями обусловлен способ изложения материала: каждый метод и алгоритм читаются независимо от изложения других методов. Для этого каждый метод вместе с соответствующим алгоритмом описывается в отдельном параграфе, который имеет единую «стандартную» структуру и состоит из: формулировки решаемой задачи, предположений (ограничивающих применяемость предлагаемого алгоритма), краткого описания идеи метода, полного текста детально разработанного алгоритма, теорем сходимости (дающих теоретическое обоснование для построенных алгоритмов), замечаний и практических рекомендаций по использованию данного алгоритма.

При написании пособия авторы сочли более целесообразным вместо готовых реализаций алгоритмов в виде конкретных программ привести их описание, так как, во-первых, любая программа — это всего лишь одна из многих реализаций исходного алгоритма, отражающая в большой степени умение программиста-математика использовать существующие возможности выбранного языка программирования с учетом реальных возможностей вычислительной техники и специфики конкретных задач, во-вторых, цель пособия — оказать помощь студентам *при составлении* программ и в работе над литературой по методам оптимизации.

Читателя не должна смущать недостаточная полнота комментариев в тексте алгоритмов. Она компенсируется достаточными библиографическими указаниями в конце каждого параграфа.

Список литературы включает лишь те работы, которые непосредственно использовались авторами при разработке алгоритмов.

Книга состоит из шести глав.

В первой главе описаны 20 алгоритмов одномерной оптимизации, во второй — более 40 алгоритмов безусловной минимизации дифференцируемых функций, в третьей — более 40 алгоритмов минимизации недифференцируемых функций и отыскания седловых точек.

В четвертой главе описано около 30 алгоритмов решения задач линейного программирования, пятая — содержит более 100 алгоритмов решения задач нелинейного и стохастического программирования, а шестая — около 20 алгоритмов решения минимаксных задач. Так как почти все алгоритмы сопровождаются теоретическим обоснованием в виде соответствующих теорем, то предлагаемое пособие будет полезным и для специалистов по теории оптимизации.

В силу объективных причин далеко не все результаты по численным методам оптимизации нашли отражение в данном пособии. Однако обширный охват материала и выбранный способ изложения позволяют надеяться, что предлагаемая книга послужит хорошим путеводителем по современным публикациям в данной области, облегчит студентам изучение и усвоение результатов цитированных публикаций, поможет преподавателям ускорить процесс обучения студентов навыкам решения практических задач на ЭВМ.

Авторы выражают глубокую благодарность академику АН УССР В. С. Михалевичу, профессору Н. З. Шору, заведующему кафедрой теории управления Львовского госуниверситета М. Я. Бартишу и сотрудникам кафедры моделирования сложных систем Киевского госуниверситета за ценные предложения по улучшению пособия.

ВВЕДЕНИЕ

ЭЛЕМЕНТЫ ТЕОРИИ ОПТИМИЗАЦИИ И УПРАВЛЕНИЯ

0.1. Математические методы управления и задачи оптимизации

1. О математических методах поиска оптимальных решений

Идеи оптимизации и оптимального управления за последние 30—40 лет глубоко проникли во многие области научных исследований, в инженерно-конструкторские разработки, в различные сферы управления социально-экономическими процессами. Этому способствовали два благоприятствующих фактора. Во-первых, подавляющее большинство самых разнообразных задач управления и оптимизации приводятся к «стандартной» математической задаче оптимизации: *найти среди элементов* x *из заданного множества* X *тот элемент* $x^* \in X$, *который доставляет наименьшее значение* $f_0(x^*)$ *заданной функции* f_0. Во-вторых, усилия математиков увенчались за эти годы разработкой многих эффективных алгоритмов решения математических задач оптимизации на ЭВМ для широких классов множеств X и функций f_0.

Поэтому исследователю в любой конкретной области науки и техники часто достаточно перевести свою конкретную задачу управления (оптимизации) на математический язык, т. е. сформулировать ее в виде математической задачи оптимизации, для которой уже известен алгоритм решения, и затем, воспользовавшись этим готовым алгоритмом, вычислить на ЭВМ искомое *оптимальное решение* x^*, т. е. тот элемент из множества X, который удовлетворяет неравенствам

$$f_0(x^*) \leqslant f_0(x), \quad \forall x \in X. \tag{0.1}$$

Например, весьма близкими и часто легко приводимыми к математической задаче оптимизации оказываются те повседневные задачи, которые формулируются в виде: какое из возможных действий (далее перечисляются все эти возможные конкретные действия) следует предпринять и осуществить, чтобы обеспечить достижение заданной цели (далее дается конкретная формулировка преследуемой цели)?

Чтобы подобную практическую задачу привести к виду математической задачи оптимизации, необходимо, во-первых, преследуемую цель выбрать *конкретно*. Для этого потребуется выбрать некоторый важный количественный показатель y_0, минимизация (или максимизация) которого наиболее соответствовала бы целям разумного управления. К сожалению, единой методики по выявлению и выбору «наиболее важного» y_0 не существует. Более того, аксиомой важной математической теории коллективных решений и конфликтов [69, 259, 260] является предположение, что существует

столько же методов конкретизации целей, сколько ответственных управляющих с их личным опытом, умением и навыками.

Во-вторых, необходимо позаботиться и о том, чтобы выбранный показатель y_0 являлся прогнозируемым. Для этого требуется построить определенную *математическую модель* управляемого процесса, т. е. сперва перечисленные возможные действия описать с помощью элементов x некоторого множества X, а затем определить на X такую вычислимую (возможно, алгоритмически) функцию f_0, значения которой $f_0(x)$ могли бы служить надежным прогнозом для ожидаемых значений y_0 при выборе тех или иных возможных действий x. Если f_0 дает для y_0 надежный прогноз, то построенную математическую модель $y_0 = f_0(x)$ называют *y_0-адекватной*. И, наконец, построенная математическая модель должна быть достаточно простой, чтобы имелись реальные возможности для вычисления искомого решения x^* задачи (0.1) в требуемые сроки и с помощью имеющейся вычислительной техники. Такую математическую модель называют *реализуемой*. Трудные вопросы практического построения y_0-адекватных и одновременно реализуемых моделей исследовались в работах [22, 23, 26, 75, 260, 336, 242, 227, 208, 177, 178, 330, 355, 356, 228].

Таким образом, специалист, желающий решить свою практическую задачу с помощью математических методов оптимизации, должен располагать некоторым запасом реализуемых моделей, среди которых может выбирать y_0-адекватную модель. С этой целью данное пособие, во-первых, знакомит читателей с широким кругом тех задач оптимизации, для которых уже разработаны теоретически обоснованные методы их решения и, во-вторых, доводит все эти методы до детально разработанных алгоритмов, чтобы облегчить их практическое использование.

2. Основные задачи оптимизации

Обычно задачи оптимизации классифицируют по тем свойствам допустимого множества X и минимизируемой функции f, которые обеспечивают или облегчают практическое отыскание оптимального решения.

В теории оптимизации выделяют три основных класса задач: *задачи безусловной оптимизации*, или *задачи без ограничений* (в этих задачах не накладываются ограничения на допустимые значения переменной x, т. е. множество X совпадает со всем пространством $\mathcal{X}$ определения переменной x); *задачи условной оптимизации*, или *задачи с ограничениями* (в этих задачах $X \neq \mathcal{X}$) и *задачи оптимизации при неполных данных;* в этих задачах f и (или) X зависят от некоторого параметра p (числового, векторного или функционального), значение которого не полностью определено в момент выбора решения x.

В дальнейшем будем обозначать через $\arg\min\limits_{x \in X} f(x)$ оптималь-

ное решение x^* задачи (0 1), т. е. тот элемент x^* из множества X, который доставляет функции f наименьшее значение на множестве X, а множество всех оптимальных решений — через $\mathrm{Arg}\min_{x \in X} f(x)$. Это множество может состоять из одного или многих элементов, или может оказаться пустым (в этом случае задача оптимизации не имеет решения) и даже может совпадать со всем множеством X — все зависит от свойств конкретно выбранной функции f и множества X.

Множество X принято называть *допустимым множеством*, или *множеством допустимых решений*, а функцию f — *минимизируемой функцией*, или *целевой функцией*, или *функцией затрат*. В некоторых случаях требуется найти элемент x^{**}, максимизирующий функцию f на множестве X, т. е. удовлетворяющий условию

$$f(x^{**}) \geqslant f(x) \ \forall \ x \in X$$

(например, если значением $f(x)$ оценивают «прибыль» от использования решения x). Такое оптимальное решение x^{**} обозначается через $\arg\max_{x \in X} f(x)$, а множество всех оптимальных решений — через $\mathrm{Arg}\max_{x \in X} f(x)$. Функцию f называют соответственно *максимизируемой (функцией полезности, критерием качества* и др.). Очевидно, задача максимизации переходит в задачу минимизации (0.1), если умножить функцию f на (—1).

В настоящее время наиболее изученными являются **задачи линейного программирования** (в этих задачах функция f линейная, а множество X задается системами линейных равенств и (или) неравенств) и **задачи выпуклого программирования** (в этих задачах f и X выпуклы). Для решения этих задач (хороших в том смысле, что все их локальные решения $\bar{x}^*$, удовлетворяющие неравенствам

$$f(\bar{x}^*) \leqslant f(x) \quad (0.2)$$

для всех $x \in X$ из окрестности точки $\bar{x}^*$, являются одновременно искомыми оптимальными решениями x^*) разработаны довольно эффективные численные алгоритмы.

Невыпуклые и многоэкстремальные задачи, а также задачи дискретной оптимизации (с дискретным или частично дискретным множеством X) являются наиболее трудными и до настоящего времени еще не найдены эффективные алгоритмы их решения (за исключением частных случаев).

Весьма трудными являются и задачи оптимизации при неполных данных. Действительно, если значение параметра неопределенности $p \in P$ неизвестно до момента выбора решения x, то оптимальным решением «надо было бы назвать» тот элемент x^*, который для всех возможных $p \in P$ и $x \in X$ удовлетворяет неравенствам

$$f(x^*, p) \leqslant f(x, p). \quad (0.3)$$

Но это чрезмерно нереалистичное требование — таких x^* обычно не существует. Если же значение параметра p известно перед выбором элемента x, то оптимальным решением x^* является любой элемент $\varphi(p)$ из множества $\operatorname{Arg} \min\limits_{x \in X(p)} f(x, p)$,

$$\varphi(p) \in \operatorname{Arg} \min_{x \in X(p)} f(x, p). \quad (0.4)$$

Задачу отыскания функции φ из условия (0.4) называют **задачей параметрической оптимизации.**

Так как в задаче оптимизации при неполных данных p не определено до момента выбора решения x^*, то часто, особенно в экономических исследованиях, неопределенные параметры заменяют некоторыми средними значениями $\bar{p}$, $\bar{p} = \int_P p dp / \int_P dp$, и в качестве x^* выбирают $\varphi(\bar{p})$.

Более объективный учет неопределенности можно осуществить в **задаче стохастической оптимизации:** *найти элемент x^*, минимизирующий функцию*

$$f_0(x) \triangleq \int_P f(x, p) d\mu(p)$$

на множестве

$$X_\varepsilon = \{x \mid P(x \in X(p)) \geqslant 1 - \varepsilon\}.$$

Здесь $P(A)$ — вероятность события A, $\varepsilon \in (0, 1)$. В теории стохастической оптимизации построены численные алгоритмы отыскания x^* также и для тех особенно трудных и важных в приложениях случаев, когда мера μ неизвестна, а имеется лишь возможность наблюдать реализации p непосредственно в процессе реализации алгоритма. Такого рода алгоритмы адаптивной оптимизации «в реальном масштабе времени» находят все более широкое применение.

Однако ориентация на «среднее» не всегда эффективна и может в процессах управления приводить к ошибочным рекомендациям и решениям. Поэтому на практике иногда отыскивают предельные решения: «оптимистическое»

$$x_о^* = \arg \min_{x \in X} \min_{p \in P} f(x, p); \quad (0.5)$$

и «пессимистическое»

$$x_п^* = \arg \min_{x \in X} \max_{p \in P} f(x, p). \quad (0.6)$$

Первое решение $x_о^*$ рассчитано на тот благоприятный случай, когда неопределенный параметр p примет значение $\bar{p}$, равное

$$p_о^* = \arg \min_{p \in P} f(x_о^*, p) \quad (0.7)$$

и этим обеспечит наилучшее возможное значение $f(x_о^*, p_о^*)$, удовлетворяющее неравенству

$$\forall x \in X \quad \forall p \in P \quad f(x_о^*, p_о^*) \leqslant f(x, p). \quad (0.8)$$

Достоинством пессимистического решения $x_п^*$ является наличие гарантии — какое бы значение $\bar{p}$ не принял параметр p, полученное реальное значение $f(x_п^*, \bar{p})$ всегда будет не больше числа

$$F = f(x_п^*, \arg\max_{p \in P} f(x_п^*, p)),$$

т. е.

$$\forall \bar{p} \in P; \quad f(x_п^*, \bar{p}) \leqslant F. \tag{0.9}$$

В этой связи решение $x_п^*$ называют еще *гарантирующим решением,* или *минимаксным решением* (по виду определяющей формулы (0.6)), а задачу (0.6) называют **задачей минимаксной оптимизации.** На другое важное свойство гарантирующего решения указывает предложение 1.

Предложение 1. Если $\bar{x}$ не является гарантирующим решением, то найдется такое $\bar{p} \in P$, что

$$f(\bar{x}, \bar{p}) > f(x_п^*, \bar{p}). \tag{0.10}$$

Поэтому при малых значениях F гарантирующее решение имеет практическое значение и часто находит применение. В остальных случаях приходится либо идти на риск, либо уточнять постановку задачи с целью уменьшения множества неопределенности P. Возможности уменьшения неопределенности p для повышения эффективности принимаемых решений изучает также теория конфликта и коллективных решений, которая основана на предположении, что значение параметра p выбирает некоторый «заинтересованный участник конфликта (второй игрок)», стремящийся минимизировать «свою целевую функцию» $f_2(x, p^*)$. Здесь и возникает ряд новых конкретных постановок задач оптимизации в зависимости от «порядка ходов», разнообразия возможностей взаимной информированности сторон как о множествах X, P, так и о целевых функциях f, f_2. Полагают, что «второй игрок делает первый ход», если значение p выбирается раньше, чем значение x. Аналогично — «первый игрок делает первый ход», если значение x выбирается раньше значения p. Заметим, что разнообразие практических задач управления служит неисчерпаемым источником разнообразных конкретных задач оптимизации в конфликтных ситуациях [33, 69, 71, 50, 201, 206, 260]. Так, если в случае полной взаимной информированности о функциях f, f_2 и о множествах X, P первый ход делает второй игрок и, следовательно, становится известным его выбор p, то мы оказываемся в условиях полной информированности и поэтому можем воспользоваться решением задачи параметрической оптимизации (0.4), сформулированной выше, т. е. оптимальным решением будет $x^* \in \varphi(p)$. Но ведь и второй игрок тоже может решить задачу (0.4) и отыскать «наше» φ. Это позволит ему заблаговременно вычислить значение $f_2(\varphi(p), p)$ для каждого значения p, а затем

выбрать свое наилучшее значение

$$p^* \in \operatorname{Arg} \min_{p \in P} \max_{x \in \varphi(p)} f_2(x, p) \triangleq \bar{P}.$$

Если и мы проделаем такие же вычисления, то также найдем это «более узкое» множество неопределенности $\bar{P} \subset P$. А в случае единственных значений φ и p^* целевая функция f примет значение $f(\varphi(p^*), p^*)$ заведомо лучшее, чем гарантированное значение F, т. е. $f(\varphi(p^*), p^*) < F$ (и только в редких случаях это неравенство превратится в равенство).

Заметим, что теория оптимизации в конфликтных ситуациях изучает способы отыскания «лучших», чем $f(\varphi(p^*), p^*)$, решений. Так, если «навязать» решение $x = \varphi(p)$ в ответ на выбор p, то, как и выше, найдем «более узкое» множество неопределенности

$$\bar{P}(\bar{\varphi}) \triangleq \operatorname{Arg} \min_{p \in P} \max_{x \in \varphi(p)} f_2(x, p),$$

зависящее от $\bar{\varphi}$. Это позволит найти и «оптимальное» $\bar{\varphi}^*$, удовлетворяющее условию

$$\bar{\varphi}^* \in \operatorname{Arg} \min_{\bar{\varphi}} \max_{p \in \bar{P}(\bar{\varphi})} f(\bar{\varphi}(p), p).$$

Важный класс задач оптимизации при неполных данных составляют **задачи многоэтапной оптимизации (задачи коррекции)**, возникающие в тех случаях, когда имеется возможность перенести на более поздние сроки выбор той или иной части искомого решения x. Например, задача двухэтапной оптимизации возникает при появлении возможности отложить на более поздний срок выбор некоторой определенной части x_2 искомого решения $x = (x_1, x_2) \in X$, т. е. сперва выбираем x_1 (1-й этап), а выбор x_2 (2-й этап) из $X(x_1)$ (сечение множества X по выбранному x_1) можем отложить до момента, когда станет известной некоторая дополнительная информация о множестве неопределенности $P(x)$ (т. е. вместо первоначального $P(x)$ станет известно некоторое «более узкое» множество $P_1(x) \subset P(x)$). Тогда целесообразно выбрать x_1 на первом этапе как оптимальное решение x_1^* для следующей вспомогательной задачи оптимизации при неполных данных: минимизировать по x_1 функцию $f(x_1, p_1, p)$ с неопределенным параметром $(p_1, p) \in P_2(x_1) \times P(x_1, X(x_1))$, где $P_2(x_1)$ — множество решений $x_2^* \in X(x_1)$ задачи минимизации по x_2 функции $f(x_1, x_2, p)$ с заданными x_1 и неопределенным параметром $p \in P(x)$.

Другая модификация задачи двухэтапной оптимизации возникает в тех случаях, когда функцию полезности $f(x_1, x_2, p)$ выбирают таким образом, чтобы учесть издержки как на получение дополнительной информации о множестве $P(x)$, так и связанные с отстрочкой выбора x_2.

3. О задачах многоцелевого управления

Трудности приведения многочисленных пожеланий и устремлений к некоторой «единой цели» зачастую оказываются непреодолимыми и тогда, не имея возможности сформулировать задачу оптимизации, мы вынуждены формулировать следующую недостаточно строгую **задачу многоцелевого управления:** *какие из возможных действий следует осуществить, чтобы обеспечить достижение следующих целей (далее перечисляются все цели)?*

Перечисляемые цели могут выражать требования: минимизировать определенные показатели

$$y_0, y_1, \ldots, y_m, \tag{0.11}$$

максимизировать некоторые другие показатели

$$y_{m+1}, \ldots, y_k, \tag{0.12}$$

удовлетворить равенствам и (или) неравенствам

$$\begin{gathered} y_{k+1} = 0, \ \ldots, \ y_s = 0; \\ y_{s+1} \leqslant d_{s+1}, \ \ldots, \ y_n \leqslant d_n. \end{gathered} \tag{0.13}$$

Если для данной практической задачи построить математическую модель f_i, $i = 0, 1, \ldots, n$, дающую надежный прогноз $f_i(x)$ величинам y_i при выборе тех или иных возможных действий $x \in X$, то эту задачу можно переформулировать как задачу отыскания среди элементов x из множества X того элемента x^*, который удовлетворяет условиям

$$\forall x \in X \ f_i(x^*) \leqslant f_i(x), \quad 0 \leqslant i \leqslant m; \tag{0.14}$$

$$\forall x \in X \ f_i(x^*) \geqslant f_i(x), \quad m < i \leqslant k; \tag{0.15}$$

$$f_i(x^*) = 0, \quad k < i \leqslant s; \tag{0.16}$$

$$f_i(x^*) \leqslant \alpha_i, \quad s < i \leqslant n. \tag{0.17}$$

К сожалению, такая задача оказывается бессодержательной, так как решения x^*, удовлетворяющего противоречивым условиям (0.14) — (0.17), обычно не существует (и тем не менее иногда встречаются подобные формулировки цели в виде «максимизировать блага (например, выпуск продукции) при минимальных затратах»).

Для того чтобы «довести» задачу многоцелевого управления до содержательной математической задачи оптимизации можно оставить некоторое наиболее важное среди неравенств (0.14)—(0.15), а остальные привести к виду (0.17) с помощью вспомогательных констант d_i, $i \leqslant k$. Выбирая различные константы d_i, исследователь может «осмотреть» соответствующие оптимальные решения $x^*(d)$ и затем выбрать «наилучшее среди них» по своим неформальным критериям.

В общем случае естественно задать такое множество B_3 и функцию B_2, чтобы условие $x \in X(B) \triangleq \{x \mid B_2(x, f_1(x), \ldots, f_n(x)) \subset \subset B_3\}$ определяло множество «удовлетворительных решений» x, а

затем с помощью вспомогательного критерия B_1 (более высокого уровня) определить *B-оптимальное решение* $x^*(B) = \arg\min_{x \in X(B)} B_1(x, f_1(x), \ldots, f_n(x))$ при условиях (0.16) — (0.17). Если, например, выбрать $B_1(x, f_1(x), \ldots, f_n(x)) \triangleq \sum_{i=0}^{m} b_i f_i(x) - \sum_{i=m+1}^{k} b_i f_i(x)$, $B_3 = \bigcup \{x\}$, то множество $\overline{X}^*$ всех $x^*(B)$, полученных при всевозможных $b_i \geqslant 0$, $\sum_{i=0}^{k} b_i = 1$ обладает замечательным свойством: в множестве X не найдется элемента z, который удовлетворял бы неравенствам $\forall i = \overline{0,\ m}$ $f_i(z) \leqslant f_i(x^*(B))$, $\forall i = \overline{m+1,\ k}$ $f_i(z) \geqslant f_i(x^*(B))$ с хотя бы одним сильным неравенством. В связи с этим множество $\overline{X}^*$, которое называют *множеством компромиссов,* естественно называть *решением задачи многоцелевого управления.*

На практике $\overline{X}^*$ используется лишь как один из этапов к окончательной формулировке задачи оптимизации. Окончательный выбор оптимальных значений $b_i = b_i^*$ осуществляют либо с помощью максимизации некоторого критерия $F(x^*(B), B)$ «более высокого уровня», т. е. выбирают $b^* = \arg\min_{b_i \geqslant 0, \Sigma b_i = 1} F(x^*(B), B)$, либо (в конфликтной ситуации) по договоренности «i-го игрока (стремящегося максимизировать $f_i(x)$)» с остальными. В последнем случае к существенным факторам прибавляется фактор времени t — момент принятия окончательного решения о согласованном выборе значений b. Этот фактор, а также и другие существенные факторы τ конфликта, позволяет каждой стороне выдвигать с помощью функции β_i конструктивные предложения (*β-уступки*) в виде: — «если выбор b будет завершен до момента времени t, то i-й игрок согласен с любым значением b из множества $\beta_i(t, \tau)$». Такие β-уступки помогают прийти к взаимо согласованному выбору значений b_i в тот момент t_1, когда множество $\bigcap_i \beta_i(t_1, \tau)$ окажется непусто.

Для более конкретных ситуаций исследовались многие важные постановки «коллективных» задач многоцелевого управления, когда разные цели порождаются разными участниками (организациями), и предложены разумные конкретизации коллективных решений, а также алгоритмы для их вычисления [33, 50, 69, 70, 74, 171, 201, 203, 206, 210, 257, 259, 260, 389].

0.2. О некоторых методах решения задач оптимизации

Любой алгоритм отыскания оптимального решения x^* для задачи минимизации функции f на множестве X,

$$f(x^*) \leqslant f(x) \quad \forall x \in X, \tag{0.18}$$

обычно ориентирован на решение определенного класса задач оптимизации с определенными свойствами функции f и множества X.

Поэтому более универсальные алгоритмы, ориентированные на решение более широких классов задач, обычно уступают по эффективности специализированным алгоритмам, использующим специфические свойства конкретно решаемой задачи. Этим и объясняется современное непрерывно возрастающее разнообразие алгоритмов оптимизации.

Приведем примеры основных оптимизационных методов.

1. Метод полного перебора

Если множество допустимых решений состоит из N элементов $x^1, x^2, \ldots, x^N$ и N настолько мало, что имеется практическая возможность вычислить все значения $f(x^1), f(x^2), \ldots, f(x^N)$, то сравнением полученных значений найдем наименьшее значение $f(x^{i_*})$ и одновременно искомое оптимальное решение

$$x^* = x^{i_*} = \arg\min_{i \in \overline{1,N}} f(x^i).$$

Такой метод *полного перебора* используется довольно редко, потому что множество X обычно содержит слишком много элементов и практически невозможно вычислить $f(x)$ для каждого $x \in X$.

2. О приближенных методах и оптимальных алгоритмах

Среди приближенных методов оптимизации наиболее часто используются:

Методы частичного перебора и случайного поиска. В этих методах выбирают некоторое достаточно представительное подмножество $X_N = \{x^1, x^2, \ldots, x^N\} \subset X$, допускающее возможности использовать метод полного перебора и найденное x^{i_*} принимают в качестве приближенного решения исходной задачи оптимизации. Если элементы $x^1, x^2, \ldots$ вычислять как реализации некоторой случайной величины $\xi \in X$ с заданной мерой μ, $\int_X d\mu(x) = 1$ (такой метод называют *методом случайного поиска,* или *методом Монте — Карло*) или задать в виде некоторой сетки узлов $X_N \subset X$, то получим метод *пассивного поиска.* В методах *активного* (или *адаптивного*) *поиска* элементы $x^1, x^2, \ldots$ вычисляют последовательно и, учитывая текущую информацию о значениях $f(x^1), f(x^2), \ldots$ (или производных f), стремятся обеспечить более густую сетку (увеличить значения $\mu(x)$) в тех подмножествах $X^N \subset X$ (с меньшими предсказуемыми значениями $f(x)$), которые становятся более подозрительными (в процессе вычислений $f(x^i)$) «на оптимум» (т. е. на то, что $X^* \subset X^N$).

Методы отсечений. В этих методах находят и «отсекают» от множества X те его подмножества $\hat{X}$, которые заведомо не содержат оптимальных решений, и получают эквивалентную задачу минимизации функции f на «меньшем» множестве $\check{X} = X \setminus \hat{X}$. Последова-

тельными отсечениями иногда удается получить последовательность таких вложенных подмножеств $\check{X}_1 \supset \check{X}_2 \supset \check{X}_3 \supset \ldots \supset \check{X}_k \supset \supset \ldots$, для которых $\operatorname{diam}(\check{X}_k) \underset{k\to\infty}{\to} 0$ (тогда при достижении неравенства $\operatorname{diam}(\check{X}_k) \leqslant \varepsilon$, любой элемент $\bar{x}_k \in \check{X}_k$ очевидно удовлетворит неравенству $\| \bar{x}_k - x^* \| \leqslant \varepsilon$). Различные эффективные варианты методов отсечений разрабатываются в рамках методов ветвей и границ, методов минорант, методов эллипсоидов (особенно эффективных для решения выпуклых задач) и других методов.

Методы аппроксимаций. В этих методах функцию f и (или) множество X заменяют некоторыми реализуемыми аппроксимациями $\tilde{f}$, $\tilde{X}$; допускающими практическую возможность вычислить $\tilde{x}^* = \arg\min\limits_{x\in\tilde{X}} \tilde{f}(x)$. Если $(\tilde{f}, \tilde{X})$ достаточно близки к (f, X), то можно ожидать, что и $\tilde{X}^*$ будет близким к $\operatorname{Arg}\min\limits_{x\in X} f(x)$.

Методы локальной оптимизации. Если для заданных $x^k \in X$, $\lambda_k > 0$ множество X заменять «меньшим» множеством $X_1(x^k, \lambda_k) = \{x \in X \mid \| x^k - x \| \leqslant \lambda_k\}$ и последовательно вычислять

$$x^{k+1} = \arg\min_{x\in X_1(x^k, \lambda_k)} f(x), \qquad (0.19)$$

то при весьма общих условиях получим сходимость $x^k \to \bar{X}^*$ к *локальному решению* задачи оптимизации, т. е. к такому $\bar{x}^* \in X$, которое при некотором $\lambda > 0$ удовлетворяет условию $\bar{x}^* = \arg\min\limits_{x\in X_1(\bar{x}^*, \lambda)} f(x)$ (очевидно при $x^{k+1} = x^k$, точка x^k является локальным решением). Поиск локальных решений оказывается особенно эффективным при решении тех задач оптимизации, в которых все локальные решения являются одновременно оптимальными решениями (например, выпуклых задач оптимизации).

Методы последовательных приближений. Так как вспомогательная задача (0.19) может оказаться довольно трудной даже при малых λ_k, то в многочисленных методах последовательных приближений предлагается вычислять x^{k+1} как приближенное решение задачи

$$x^{k+1} = \arg\min_{x\in X^k} F_k(x), \qquad (0.20)$$

где аппроксимации F_k, X^k выбираются из соображений упростить вычисление x^{k+1}, сохранив при этом сходимость $x^k \underset{k\to\infty}{\to} \bar{x}^*$.

В качестве X^k часто выбирают некоторые из множеств X, $\bar{X}$, $X_1(x^k, \lambda_k)$, $X_2(x^k, \lambda_k) \triangleq \{x \mid \| x - x^k \| \leqslant \lambda_k\}$, $X_k(x^k, \lambda_k, J) \triangleq \triangleq \{x \mid | x_i - x_i^k | \leqslant \lambda_k,\ i \in J \subset \{1, 2, \ldots, n\}\}$ и др., или опреде-

ленные пересечения этих множеств. Очевидно, при уменьшении λ_k необходимо повышать точность аппроксимации $\nabla f(x)$, поэтому в качестве F_k обычно выбирают линейную аппроксимацию f в окрестности x^k $F_1(x^k, x) \triangleq f(x^k) + (\nabla f(x^k), x - x^k)$, реже квадратичную аппроксимацию $F_2(x^k, x) \triangleq f(x^k) + (\nabla f(x^k), x - x^k) + \frac{1}{2}(\nabla^2_{xx} f(x^k)(x - x^k), x - x^k)$ или аппроксимации более высоких порядков, часто, особенно для негладких f, выбирают негладкие аппроксимации $F_3(x^k, x) = \sum_i a_i(x^k) \max_{j \in I(x^k, i)} (c^{ij}(x^k), x)$ или композиции таких функций.

Множество возможных способов выбора последовательностей $\{F_k, X^k\}_{k=1}^{\infty}$, а также процедур P_k для приближенного вычисления решений x^{k+1} соответствующих упрощенных задач оптимизации (0.20) порождает множество $\tilde{A}$ алгоритмов A конкретной реализации методов последовательных приближений. В связи с этим актуальной является задача выбора наиболее эффективного (оптимального) на классе $(\tilde{F}, \tilde{X})$ алгоритма $A \triangleq \{F_k, X_k, P_k, k = \overline{1, N(A)}, x^1 \in X\} \in \tilde{A}$ для вычисления по формулам (0.20) приближения $x^*(A) \triangleq x^{N(A)}$ к искомому решению $x^* = \arg\min_{x \in X} f(x)$ при условии, что f и X принадлежат заданным классам $\tilde{F}$ и $\tilde{X}$. Очевидно, это задача многоцелевой оптимизации, преследующая среди наиболее важных следующие цели [177]:

1) минимизировать время $T(A) \triangleq y_1^1(A, f, X)$ реализации алгоритма A, т. е. время, затраченное на вычисление элемента $x^*(A)$;

2) минимизировать погрешность вычисленного приближения $x^*(A)$, т. е. расстояние $\rho(A) \triangleq y_2^1(A, f, X)$ от $x^*(A)$ до множества всех x^* для функции f и множества X;

3) минимизировать $f(x^*(A)) - \min_{x \in X} f(x) \triangleq y_3^1(A, f, X)$;

4) минимизировать число итераций $N(A) = y_4^1(A, f, X)$;

5) минимизировать объем памяти $V_i(A) \triangleq y_i^2(A, f, X)$, занимаемый на i-м устройстве вычислительной системы при реализации алгоритма A;

6) минимизировать время $T_i(A) \triangleq y_i^3(A, f, X)$ работы i-го устройства при реализации алгоритма A;

7) удовлетворить требуемым ограничениям на допустимые значения $y_{\mathcal{K}}^I(A, f, X) \triangleq \{y \mid y = y_k^i(A, f, X), i = 1, 2, 3; k \in \mathcal{K}(i)\}$.

В связи с этим *B — оптимальный на классе* $(\tilde{F}, \tilde{X})$ алгоритм $A(B, \tilde{F}, \tilde{X})$ определяется, с помощью вспомогательных функций B_1, B_2 и множества B_3, как решение задачи минимизации по $A \in \tilde{A}$ заданной функции $\max_{(f,X) \in (\tilde{F}, \tilde{X})} B_1(A, y_{\mathcal{K}}^I(A, f, X))$ при заданных

ограничениях $B_2(A, y^I_{\mathcal{K}}(A, f, X)) \in B_3$, т. е.

$$A(B, \tilde{F}, \tilde{X}) = \arg \min_{A \in \tilde{A}, B_2(A, y^I_{\mathcal{K}}(A, f, X)) \in B_3} \max_{(f, X) \in (\tilde{F}, \tilde{X})} B_1(A, y^I_{\mathcal{K}}(A, f, X)).$$

Следует отметить, что если для класса $(\tilde{F}_1, \tilde{X}_1)$ построен оптимальный алгоритм $A_1 = A(B, \tilde{F}_1, \tilde{X}_1)$ и затем оптимальные алгоритмы A_2 и A_3 соответственно для более узких классов $(\tilde{F}_2, \tilde{X}_2)$ и $(\tilde{F}_3, \tilde{X}_3)$, удовлетворяющих условию $(\tilde{F}_2, \tilde{X}_2) \cup (\tilde{F}_3, \tilde{X}_3) = (\tilde{F}_1, \tilde{X}_1)$, то это еще не значит, что алгоритм A_1 уже не нужен и можно заменить его построенными более эффективными алгоритмами A_2 и A_3. Очевидно, такая замена оправдана лишь в тех случаях, когда повышение эффективности A_2 и A_3 по сравнению с A_1 компенсирует дополнительные затраты на проверку принадлежности $(f, X) \subset \subset (\tilde{F}_i, \tilde{X}_i)$ при выборе требуемого алгоритма A_i, $i = 2, 3$. Однако проверка принадлежности практической задачи (f, X) определенному классу задач $(\tilde{F}_i, \tilde{X}_i)$ зачастую оказывается почти неосуществимой и поэтому «ранжировка по эффективности» существующих алгоритмов также оказывается весьма проблематичной. В большинстве случаев правдоподобную оценку эффективности алгоритма дает лишь вычислительный эксперимент.

0.3. Методы отсечений

Приведем примеры успешного применения методов отсечений для решения ряда важных задач оптимизации.

1. Минимизация квазивыпуклых функций

Функцию $f: [a, b] \to \boldsymbol{R}$, $[a, b] \subset \boldsymbol{R}$, называют квазивыпуклой, если для любых $a \leqslant x_1 < x_2 \leqslant b$ неравенство $f(x_1) \leqslant f(x_2)$ влечет неравенство $x^* = \arg \min_{x \in [a,b]} f(x) \leqslant x_2$; а неравенство $f(x_1) \geqslant f(x_2)$ влечет неравенство $x^* \geqslant x_1$. Поэтому, выбрав $x_1 = (a + b - \varepsilon)/2$ и $x_2 = (a + b + \varepsilon)/2$, в случае $f(x_1) \geqslant f(x_2)$ получим $x^* \subset \check{X}^1 = [x_1, b]$, а в случае $f(x_1) \geqslant f(x_2)$ получим $x^* \subset \check{X}^1 = [a, x_2]$, т. е. в обоих случаях вместо $[a, b]$ найдем новый (почти вдвое короче) отрезок $\check{X}^1$ локализации x^*. Повторяя этот процесс, будем находить все более короткие отрезки локализации решения $[a, b] \supset \check{X}^1 \supset \supset \check{X}^2 \supset \ldots$

Более эффективные и даже оптимальные алгоритмы локализации x^* (по методу Фибоначчи и золотого сечения и другие) описаны в § 1.1—1.4.

2. Метод минорант. Минимизация липшицевых функций

Утверждение 1. Если функция F является минорантной для f на множестве X, т. е. для всех $x \in X$, $F(x) \leqslant f(x)$, $\bar{x} \in X$, то

$$x^* = \arg\min_{x \in X} f(x) \in \check{X}_F \triangleq \{x \mid F(x) \leqslant f(\bar{x})\} \cap X,$$

$$f(x^*) \in [\inf_{x \in X} F(x),\ f(\bar{x})].$$

Действительно, для всех $x \notin \check{X}_F$ имеем $f(x) \geqslant F(x) > f(\bar{x})$.

Следствие 1. Если

$$X_N = \{x^1,\ x^2,\ \ldots,\ x^N\} \subset X,\ M_N = \min_{i=\overline{1,N}} f(x^i) = f(x^{i^*}),$$

$$C_N \leqslant \min_{x \in X} F(x),$$

то

$$f(x^*) \in [C_N,\ M_N].$$

Поэтому при достижении неравенства $M_N - C_N < \varepsilon$ получим $f(x^{i^*}) - f(x^*) \leqslant \varepsilon$.

Утверждение 2. Если функция f липшицева на X, т. е. существует константа Липшица L, для которой

$$|f(x^1) - f(x^2)| \leqslant L \| x^1 - x^2 \|$$

при $x^1,\ x^2 \in X$, то функция

$$F_N(x) = \max_{i \in \overline{1,N}} (f(x^i) - L \| x - x^i \|)$$

является минорантой для f. Поэтому для минимизации липшицевой функции можно использовать следующий

Алгоритм 1: выбираем произвольное $x^1 \in X$, полагаем $X_1 = \{x^1\}$, $N = 1$ и для $N = 1,\ 2,\ \ldots$ вычисляем

$$x^{N+1} = \arg\min_{x \in X} F_N(x),$$

$$X_{N+1} = X_N \cup \{x^{N+1}\}.$$

Утверждение 3. Если последовательность $x^1,\ x^2,\ \ldots$ построена с помощью Алгоритма 1 для липшицевой функции f, то

$$\lim_{N \to \infty} f(x^N) = \inf_{x \in X} f(x),$$

а если при этом X — компакт, то

$$\lim_{N \to \infty} x^N = x^*.$$

3. Минимизация выпуклых функций. Метод эллипсоидов

Утверждение 4. Если функция f выпукла и ограничена на $x \in \mathbf{R}^n$ и $\bar{x} \in X$, то существует такое $C_{\bar{x}} \in \mathbf{R}^n$, что линейная функция

$$F_{\bar{x}f}(x) \equiv (C_{\bar{x}},\ x - \bar{x}) + f(\bar{x})$$

является минорантой для f на X, т. е. для всех $x \in X$ $F_{\bar{x}f}(x) \leqslant f(x)$ и, очевидно, $F_{\bar{x}f}(\bar{x}) = f(\bar{x})$.

Следствие 1. Если $x^1 \in X$, то

$$x^* \in \check{X} \triangleq \{x \mid F_{x^1 f}(x) \leqslant f(x^1)\} \cap X.$$

Следствие 2. Если $x^1, x^2, \ldots, x^N \in X$, то

$$x^* \in \check{X}_N \triangleq X \bigcap_{i=1}^{N} \{x \mid F_{x^i f}(x) \leqslant f(x^i)\}.$$

Это позволяет исходную задачу минимизации f на X заменить задачей минимизации f на «меньшем» множестве $\check{X}_N$.

По-видимому, представляет практический интерес выбирать значения $x^1, x^2, \ldots, x^N, \ldots$ таким образом, чтобы множество $\check{X}_N$ эффективно «уменьшалось» с ростом N. Для этого целесообразно выбирать элементы x^i поближе к центру тяжести или чебышевскому центру множества $\check{X}_{i-1}$.

Строятся также (y, N)-оптимальные методы, в которых элементы $x^1, x^2, \ldots, x^N$ выбираются таким образом, чтобы обеспечить минимальное значение меры $J(\check{X}_N)$ (например, диаметра $\check{X}_N$).

Особо эффективным оказался следующий метод эллипсоидов [417]. Предположим, что известно число R_1 и точка $x^1 \in \boldsymbol{R}^n$, удовлетворяющие условию $\| x^1 - x^* \| \leqslant R_1$, т. е. x^* принадлежит шару $S(x^1, R_1)$ радиуса R_1 с центром в точке x^1. Тогда x^* принадлежит также и полушару $S_1(x^1, R_1) = S(x^1, R_1) \cap \{x \mid F_{x^1 f}(x) \leqslant f(x^1)\}$. Построим (это нетрудно) гиперэллипсоид минимального объема, содержащий $S_1(x^1, R_1)$ и затем с помощью линейного преобразования «растянем» пространство таким образом, чтобы данный гиперэллипсоид опять превратился в шар. Очевидно радиус R_2 этого нового шара уменьшится по сравнению с R_1 ($R_2 = R_1 \sqrt[3]{1 - 1/2n}$). Поэтому, повторяя данный процесс, можно локализовать x^* в шарах все меньших радиусов. В итоге получаем последовательность $\{x^k\}$, удовлетворяющую неравенствам $\| x^{k+1} - x^* \| \leqslant q \| x^k - x^* \|$ при независимом (!) от f значении $q \approx 1 - 1/2n^2 < 1$, т. е. $\lim_{k\to\infty} x^k = x^*$.

Этот метод легко приспосабливается и для минимизации выпуклой функции f на заданном выпуклом множестве $X \triangleq \{x \mid g^i(x) \leqslant 0,\ i = \overline{1, m}\}$ с выпуклыми функциями $g^i : \boldsymbol{R}^n \to \boldsymbol{R}$. Действительно, введем вспомогательную функцию $\tilde{F}(x) = f(x)$ на X и $\tilde{F}(x) = g^{i_k}(x)$ вне X ($i_k = \arg\max_i g^i(x^k)$). Так как $\operatorname{Arg}\min_{x \in X} f(x) \in \operatorname{Arg}\min_x \tilde{F}(x) = X$ и при этом $\tilde{F}$ выпукла вне X (нетрудно проверить, что $F_{x^k \tilde{F}} = F_{x^k f}$ при $x \in X$ и $F_{x^k \tilde{F}} = F_{x^k g^{i_k}}$ при $x \notin X$), то требуемый полушар можем «отсекать» с помощью $F_{x^k \tilde{F}}$.

Формально к методам отсечений можно отнести и методы локализации решений на подмножествах $\hat{X} \subset X$ тех элементов x, которые удовлетворяют необходимым условиям оптимальности. Часто необходимые условия оптимальности оказываются достаточно конструктивными и позволяют отыскать x^* (либо локализовать его в «достаточно малых» подмножествах $\check{X}$).

0.4. О локализации решений с помощью необходимых и достаточных условий оптимальности. Дополнительная терминология

1. Необходимые условия оптимальности для дифференцируемых функций

Функцию f называют дифференцируемой в точке $\bar{x}$, если существует такой вектор $\nabla f(\bar{x})$, что для всех x в окрестности $\bar{x}$ справедливо асимптотическое равенство

$$f(x) = f(\bar{x}) + (\nabla f(\bar{x}),\ x - \bar{x}) + o(\|x - \bar{x}\|).$$

Вектор $\nabla f(\bar{x})$ называют градиентом или производной функции f в точке $\bar{x}$.

Утверждение 1. Оптимальное решение x^*, минимизирующее дифференцируемую функцию f на множестве X, находится либо на границе $\bar{X}$ множества X, либо на множестве $\bar{X}^*$ решений $\bar{x}^*$ уравнения $\nabla f(x) = 0$.

Следовательно, задачу минимизации f на X можем заменить задачей минимизации f на подмножестве $\check{X} = \bar{X} \cup (\bar{X}^* \cap X) \subset X$.

Справедливость утверждения 1 следует из того, что предположение $\nabla f(x^*) \neq 0$ для $x^* \notin \bar{X}$ ведет к противочивому (при малых $\lambda > 0$) неравенству $f(x^* - \lambda\nabla f(x^*)) = f(x^*) - \lambda \|\nabla f(x^*)\|^2 + o(\lambda) < f(x^*)$. Если существует $f'(x^k, z) \triangleq \lim\limits_{\lambda\to 0+} (f(x^k + \lambda z) - f(x^k))/\lambda$, то $f'(x^k, z)$ называют производной по направлению z для функции f в точке x^k.

Утверждение 2. Если f дифференцируема в точке x^k, то $f'(x^k, z) = (\nabla f(x)^k, z)$.

Направление $z(x^k) \triangleq \arg\min\limits_{\|z\|=1} f'(x^k, z)$ называют *направлением наибыстрейшего убывания* функции f в точке x^k.

Следствие 1. Если f дифференцируема, то $z(x^k) = -\nabla f(x^k)$.

Это значит, что приближение x^{k+1}, лучшее чем x^k, можно вычислить по формуле $x^{k+1} = x^k + \lambda z(x^k)$, $\lambda = \arg\min\limits_{\lambda>0} f(x^k + \lambda z(x^k))$.

Следствие 2. Если $x^* = \arg\min_x [(Cx, x) + (d, x)]$, то x^* — решение линейной системы алгебраических уравнений $(C + C^*) x + d = 0$.

Лемма [23]. Если $\bar{x} = \arg[(C + C^*)x = d]$, $d^1 \triangleq A (C + C^*) d$, $A_1 = I - A (C + C^*)$, $d^{k+1} = d^k + A_k d^k$, $A_{k+1} = A_k^2$, то $\| \bar{x} - d^k \| \leqslant \| A_1 \|^{2^{k+1}} \| d \|$. Поэтому d^k эффективно приближается к искомому $\bar{x}$ с увеличением k.

Приведем ряд условий оптимальности, справедливых и для граничных точек множества X.

2. Необходимые и достаточные условия оптимальности для задачи выпуклого программирования. Теорема Куна—Таккера

Напомним необходимые понятия выпуклого анализа.

Пусть $x^1, x^2 \in X$. Функцию f называют *выпуклой* на выпуклом множестве X, если $\forall \lambda \in [0, 1]$ $f(x_\lambda) \leqslant \lambda f(x^1) + (1 - \lambda) f(x^2)$, называют *строго выпуклой,* если $\forall \lambda \in [0, 1]$ $f(x_\lambda) < \lambda f(x^1) + (1 - \lambda) f(x^2)$ и называют *сильно выпуклой,* если при некотором $\alpha > 0$

$$f\left(\frac{x^1 + x^2}{2}\right) \leqslant (f(x^1) + f(x^2))/2 - \alpha \| x^1 - x^2 \|,$$

$$x_\lambda \triangleq \lambda x^1 + (1 - \lambda) x^2.$$

Аналогично, множество X называют *выпуклым,* если $\forall \lambda \in [0, 1]$ $x_\lambda \in X$, называют *строго выпуклым,* если при $\lambda \in (0, 1)$ x_λ является внутренней точкой X, и называют *сильно выпуклым,* если при некотором $\alpha > 0$ открытое множество $\{x \mid \| x - x_\lambda \| \leqslant \alpha \| x^1 - x^2 \| \min \{\lambda, 1 - \lambda\}\}$ принадлежит внутренности множества X, $\lambda \in (0, 1)$.

Утверждение 2. Если функция f выпукла и дважды дифференцируема, то следующие условия эквивалентны:

$$(1)\ f(x^2) - f(x^1) \geqslant (\nabla f(x^1), x^2 - x^1);$$

$$(2)\ (\nabla f(x + \lambda h), h) \text{ — неубывающая функция } \lambda;$$

$$(3)\ (\nabla_{xx}^2 f(x) h, h) \geqslant 0 \quad \forall x, h \in \boldsymbol{R}^n.$$

Утверждение 3. Если f — дважды непрерывно дифференцируемая функция, то условие сильной выпуклости эквивалентно условию $(\nabla_{xx}^2 f(x) h, h) \geqslant \varepsilon \| h \|^2$, $\varepsilon > 0$, $x, h \in \boldsymbol{R}^n$; сильно выпуклая f достигает своего минимума на замкнутом множестве.

Естественно, выпуклая и даже строго выпуклая функция f может не достигать своего минимума на замкнутом X.

Утверждение 4. Строго выпуклая функция достигает своего минимума на выпуклом множестве X лишь в единственной точке.

Множество X^* всех решений выпукло и довольно просто описывается с помощью конуса допустимых направлений. Множество $K(x^1)$ называют *конусом допустимых направлений* для X в точке x^1,

если оно состоит из тех векторов h, для которых существует число $\varepsilon > 0$, удовлетворяющее условию $x^1 + \varepsilon h \in X$.

Утверждение 5. Если f и X выпуклы и f непрерывно дифференцируема, то необходимым и достаточным условием оптимальности точки $\bar{x}$ является включение

$$\nabla f(\bar{x}) \in K^*(\bar{x}),$$

$(K^*(\bar{x}) \triangleq \{y \mid (y, z) \geqslant 0 \quad \forall z \in K(\bar{x})\}$ — сопряженный конус), т. е.

$$X^* \triangleq \operatorname{Arg}\min_{x \in X} f(x) = \check{X} \triangleq \{x \mid \nabla f(x) \in K^*(x), \ x \in X\}.$$

Следствие 1. $X^* = \{x \mid (\nabla f(x), \ z - x) \geqslant 0 \quad \forall z \in X\} \cap \cap X$, а в случае $X = \boldsymbol{R}^n$ имеем $X^* = \{x \mid \nabla f(x) = 0\}$.

Вектор $\hat{\nabla} f(\bar{x}) \in \boldsymbol{R}^n$ называют *субградиентом*, или *обобщенным градиентом* в точке $\bar{x}$, если

$$f(x) \geqslant f(\bar{x}) + (\hat{\nabla} f(\bar{x}), \ x - \bar{x}) \ \forall x.$$

Множество $M(x)$ субградиентов непрерывной выпуклой функции f в точке x является выпуклым, замкнутым и ограниченным.

Утверждение 6. Для выпуклой функции f производная $f'(\bar{x}, z)$ по направлению z равна $\max\limits_{y \in M(\bar{x})} (z, y)$.

Следствие 1. Направление $z(\bar{x})$ наибыстрейшего убывания функции f в точке $\bar{x}$ равно

$$z(\bar{x}) \triangleq \arg\min_{\|z\|=1} f'(\bar{x}, z) = \arg\min_{\|z\|=1} \max_{y \in M(\bar{x})} (z, y).$$

Это значит, что в методах последовательных приближений в качестве X^k можно выбрать множество $X^k = \{x \mid \max\limits_{y \in M(x^k)} (x, y) \leqslant \leqslant -\lambda_k \| x \|\}$ для некоторого $\lambda_k > 0$.

Следствие 2. Для того чтобы для выпуклой функции f $\bar{x} = \arg\min\limits_{x} f(x)$ необходимо и достаточно, чтобы $0 \in M(\bar{x})$.

В общей задаче выпуклого программирования множество X имеет вид $X = X^1 \triangleq \{x \mid f_i(x) \leqslant 0, \ i = \overline{1, m}, \ x \in Y\}$, где Y — заданное выпуклое множество, а f_i — выпуклые функции.

Теорема Куна — Таккера. *Для того чтобы точка $\bar{x}$ минимизировала выпуклую функцию f_0 на множестве X^1, удовлетворяющем условию Слейтера:*

$$\exists x^1 \in Y \quad \forall i = \overline{1, m} \quad f_i(x^1) < 0,$$

необходимо и достаточно, чтобы существовало решение $\bar{u}_1, \ldots, \bar{u}_m$ системы

$$f_0(\bar{x}) + \sum_{j=1}^{m} \bar{u}_j f_j(\bar{x}) \leqslant f_0(x) + \sum_{j=1}^{m} \bar{u}_j f_j(x) \quad \forall x \in Y,$$

$$\bar{u}_j f_j(\bar{x}) = 0, \quad \bar{u}_j \geqslant 0 \quad \forall j = \overline{1, m}.$$

Такое решение $\bar{u}_1, \ldots, \bar{u}_m$ называют *множителями Лагранжа*, а пару $(\bar{x}, \bar{u})$ — *точкой Куна — Таккера*. Функцию $\varphi(x, u) \triangleq f_0(x) + \sum_{j=1}^{m} u_j f_j(x)$ называют *функцией Лагранжа*. Пару $(\bar{x}, \bar{u})$ называют *седловой точкой* функции φ на множестве $Y \times R_+^n$, если $\bar{x} \in Y$, $\bar{u} \geqslant 0$ и выполнены неравенства $\varphi(\bar{x}, u) \leqslant \varphi(\bar{x}, \bar{u}) \leqslant \varphi(x, \bar{u})$ $\forall x \in Y$, $\forall u \geqslant 0$. Поэтому теорему Куна — Таккера можно сформулировать следующим образом: если выполнено условие Слейтера, то необходимым и достаточным условием оптимальности $\bar{x}$ является существование такого вектора $\bar{u} \geqslant 0$, при котором пара $(\bar{x}, \bar{u})$ является седловой точкой функции Лагранжа на множестве $Y \times R_+^n$.

Утверждение 7. Если выполнено условие Слейтера, то для оптимальности точки $\bar{x}$ (минимизирующей выпуклую функцию f_0 на выпуклом множестве X^1), необходимо и достаточно существование таких векторов $\hat{\nabla} f_i(\bar{x}) \in M_i(\bar{x})$ и чисел u_i, $i = \overline{1, m}$, что

$$u_i f_i(\bar{x}) = 0, \quad u_i \leqslant 0, \quad i = \overline{1, m},$$

$$\hat{\nabla} f_0(\bar{x}) = \sum_{i=1}^{m} u_i \hat{\nabla} f_i(\bar{x}).$$

3. Необходимые условия первого и второго порядка для задач нелинейного программирования

Утверждение 8. Для того чтобы точка $\bar{x}^*$ являлась точкой локального минимума функции f_0 на множестве $X = \{x \mid f_i(x) \leqslant 0,\ i \in \mathcal{I}^-,\ f_i(x) = 0,\ i \in \mathcal{I}^0\}$ необходимо существование таких не всех равных нулю чисел u_0, u_i, $i \in \mathcal{I}^-$, v_i, $i \in \mathcal{I}^0$, что

$$u_0 \nabla f_0(\bar{x}^*) + \sum_{i \in \mathcal{I}^-} u_i \nabla f_i(\bar{x}^*) + \sum_{i \in \mathcal{I}^0} v_i \nabla f_i(\bar{x}^*) = 0,$$

$$u_0 \geqslant 0, \quad u_i \geqslant 0, \quad i \in \mathcal{I}^-, \quad u_i f_i(\bar{x}^*) = 0, \quad i \in \mathcal{I}^-.$$

Точку локального минимума $\bar{x}^*$ называют регулярной, если линейно независимы все вектора $\nabla f_i(\bar{x}^*)$ с теми индексами i, для которых выполняются равенства $f_i(\bar{x}^*) = 0$.

Следствие 1. Если точка $\bar{x}^*$ регулярна, то можно положить $u_0 = 1$, остальные множители u_i, v_i определяются однозначно.

Следующее утверждение дает необходимые условия оптимальности второго порядка.

Утверждение 9. Если функции f_i, $i \in \mathcal{I}^0 \cup \mathcal{I}^- \cup \{0\}$, дважды непрерывно дифференцируемы и $\bar{x}$ — регулярная точка минимума, то существуют такие числа u_i, v_i, что

$$\nabla_x \varphi(\bar{x}, u, v) = 0, \quad u_i \geqslant 0, \quad u_i f_i(\bar{x}) = 0, \quad i \in \mathcal{I}^-,$$

$$(\nabla_{xx}^2 \varphi(\bar{x}, u, v) h, h) \geqslant 0$$

для всех h, удовлетворяющих условиям

$$(\nabla f_i(\bar{x}),\ h) \leqslant 0, \quad i \in \{i \mid f_i(\bar{x}) = 0, \quad u_i \leqslant 0,\ i \in \mathcal{I}^-\};$$

$$(\nabla f_i(\bar{x}),\ h) = 0, \quad i \in \{i \mid u_i > 0, \quad i \in \mathcal{I}^-\} \cup \mathcal{I}^0;$$

$$\varphi(x,\ u,\ v) \triangleq f_0(x) + \sum_{i \in \mathcal{I}^-} u_i f_i(x) + \sum_{i \in \mathcal{I}^0} v_i f_i(x).$$

4. Условия оптимальности для задач минимизации негладких функций

Простейшим примером негладкой функции является часто встречающаяся функция вида $f_0(x) = \max_{i \in \mathcal{I}} f_i(x)$, $\mathcal{I} \triangleq \{1,\ 2,\ \ldots,\ m\}$.

Утверждение 10. Если $\bar{x} = \arg\min_x \max_{i \in \mathcal{I}} f_i(x)$ для заданных дифференцируемых функций f_i, $i \in \mathcal{I}$, то существуют такие числа $u_i \geqslant 0$, $\sum_{i \in \mathcal{I}} u_i = 1$, что выполняются равенства

$$\sum_{i \in \mathcal{I}} u_i \nabla f_i(\bar{x}) = 0, \quad u_i\left(f_i(\bar{x}) - \max_{i \in \mathcal{I}} f_i(\bar{x})\right) = 0, \quad i \in \mathcal{I}.$$

Выше приводилось весьма важное свойство выпуклых функций, состоящее в том, что производная по направлению z $f'(x^1,\ z)$ равняется $\max_{y \in M(x^1)} (y,\ z)$. Это свойство положено в основу определения класса *квазидифференцируемых* функций, т. е. таких функций f, для которых в каждой точке x определено замкнутое выпуклое множество $M_f(x)$ (*множество квазиградиентов*), удовлетворяющее для всех $z \in R^n$ равенству $f'(x,\ z) = \max_{y \in M_f(x)} (y,\ z)$ [314]. Поэтому все выпуклые функции принадлежат классу квазидифференцируемых функций. Квазидифференцируемыми являются также и все функции f_0 вида $f_0(x) = \max_{y \in Y} f(x,\ y)$ с дифференцируемыми f и замкнутым ограниченным множеством Y (в данном случае $M_f(x) = \mathrm{co}\,\{z \mid z = \nabla_x f(x,\ y^*),\ y^* = \arg\max_{y \in Y} f(x,\ y)\}$), а также все слабовыпуклые функции [272] (те функции f, для которых существует непустое множество $G_f(x)$, удовлетворяющее условию $\forall h \in G_f(x)\ \forall y \in R^n\ f(y) - f(x) \geqslant (h,\ \ y - x) + r(x,\ y)$, $\frac{r(x,\ y)}{\|x - y\|} \underset{x \to y}{\to} 0$ равномерно по x в каждом ограниченном замкнутом множестве $Q \ni x$) и функции вида $f_0(x) = \max_{a \in A} f(x,\ a)$, где $f(x,\ a)$ — непрерывны по a и слабовыпуклы по x при любом фиксированном a из компактного топологического пространства A. В последнем случае $M_{f_0}(x) = \mathrm{co} \bigcup_{a \in A} G_{f(\cdot,a)}(x)$.

Утверждение 11. Для того чтобы точка $\bar{x} \in R^n$ минимизировала квазидифференцируемую функцию f, необходимо, чтобы $0 \in M_f(\bar{x})$.

Утверждение 12. Вектор $z(x^1) = \arg\min_{\|z\|=1} \max_{y \in M_f(x^1)} (y, z)$ определяет направление наибыстрейшего убывания функции f в x^1.

В работе [129] дано более общее определение квазидифференцируемой функции, а именно: функция $f: \boldsymbol{R}^n \to \boldsymbol{R}$ называется квазидифференцируемой в точке $\bar{x}$, если она дифференцируема в этой точке по направлению и если существуют такие выпуклые компакты $\underline{\partial} f(\bar{x}) \subset \boldsymbol{R}^n$ и $\bar{\partial} f(\bar{x}) \subset \boldsymbol{R}^n$, что для любого $z \in \boldsymbol{R}^n$

$$f'(\bar{x}, z) \doteq \max_{y \in \underline{\partial} f(\bar{x})} (z, y) + \min_{y \in \bar{\partial} f(\bar{x})} (z, y).$$

Множество таких функций замкнуто относительно операций взятия максимума и минимума по конечному числу функций и относительно всех «алгебраических» операций.

Теорема [129]. *Если* $\bar{x} = \arg\min_x f(x)$, то $-\bar{\partial} f(\bar{x}) \subset \underline{\partial} f(\bar{x})$; *если* $\tilde{x} = \arg\max_x f(x)$, *то* $-\underline{\partial} f(\tilde{x}) \subset \bar{\partial} f(\tilde{x})$.

Задачу условной (и параметрической) оптимизации можно привести методом развязывающей декомпозиции к более простой задаче минимизации $f_0(x, p)$ на реализуемом носителе $f(x, p) = 0$, определяющем функцию $p \to x(p)$ [22].

Утверждение 13. Квазидифференциалы $\hat{f}_0$ функции $f_0(x(p), p)$ по p в точке $\bar{p}$ вычисляются с помощью квазидифференциалов $\bar{f}_0$, $\bar{f}$ функций f_0, f по x в точке $(\bar{x}(\bar{p}), \bar{p})$ как квазидифференциалы по p в $\bar{p}$ для функций $f_0(\bar{x}, p) + z(f(\bar{x}, p), p)$ при $\bar{f}_0 \equiv \bar{z}(\bar{f}, p)$ по переменной x.

0.5. Методы последовательных приближений

Методами последовательных приближений строят такую последовательность $x^1, x^2, \ldots, x^n \in X$, которая либо является *минимизирующей,* т. е. удовлетворяет условиям

$$\lim_{n\to\infty} f(x^n) = \inf_{x \in X} f(x), \tag{0.21}$$

либо *сходится к решению,* т. е.

$$\lim_{n\to\infty} x^n = x^* = \arg\min_{x \in X} f(x), \tag{0.22}$$

либо определяет *стационарное решение,* удовлетворяющее тем или иным необходимым условиям оптимальности, например,

$$\lim_{n\to\infty} \| \nabla f(x^n) \| = 0. \tag{0.23}$$

Очевидно, сходимость (0.22) обеспечивается выполнением неравенств

$$\| x^{n+1} - x^* \| \leqslant q_n \| x^n - x^* \|$$

при $q_n \leqslant q < 1$ для всех $n = 1, 2, \ldots$. В данном случае говорят, что метод сходится со скоростью *геометрической прогрессии,* или *с линейной скоростью;* в случае $q_n \to 0$ — *со сверхлинейной скоростью,* в случае $q_n \leqslant C \| x^n - x^* \|$ — *с квадратичной скоростью,* а в случае $q_n \leqslant C \| x^n - x^* \|^{k-1}$ — *со скоростью k-го порядка.*

Способы вычисления $(n+1)$-го приближения x^{n+1} обычно основаны на использовании получаемой информации о значениях $f(x^n)$, $f(x^{n-1})$, ..., а иногда и производных первого $\nabla f(x^n)$, $\nabla f(x^{n-1})$, ..., второго $\nabla^2_{xx} f(x^n)$, $\nabla^2_{xx} f(x^{n-1})$, ... и (в редких случаях) производных более высоких порядков. Эта информация позволяет выбирать правдоподобные гипотезы о возможной конфигурации функции f (в окрестности точки x^n) и затем, на основе принятой гипотезы, определить такое значение x^{n+1}, для которого ожидается выполнение неравенства

$$f(x^{n+1}) < f(x^n) - C_n \tag{0.24}$$

с возможно большим значением числа $C_n > 0$. Выбор различных гипотез приводит к различным методам оптимизации. Если функция f непрерывно дифференцируема, то, как известно,

$$f(x) = f(x^n) + (\nabla f(x^n),\ x - x^n) + o(\| x - x^n \|). \tag{0.25}$$

Отсюда видим, что направление наибыстрейшего убывания функции f совпадает с вектором $-\nabla f(x^n)$ и поэтому можем при малых λ_n вычислять x^{n+1} по формуле:

$$x^{n+1} = x^n - \lambda_n \nabla f(x^n). \tag{0.26}$$

Метод (0.26) называют *методом градиентного спуска,* или *градиентным методом.*

Различные модификации градиентных методов основаны на том, что функция $F_1(x) \equiv f(x^n) + (\nabla f(x^n), x - x^n)$ (а значит и непрерывно дифференцируемая функция f) убывает не только в направлении $-\nabla f(x^n)$, но и по всем направлениям z^n, удовлетворяющим неравенству

$$(z^n,\ \nabla f(x^n)) \leqslant \delta < 0 \tag{0.27}$$

(действительно, $F_1(x^n + z^n) = f(x^n) + (\nabla f(x^n),\ x^n + z^n - x^n) = F_1(x^n) + (\nabla f(x^n),\ z^n) < F_1(x^n)$). Поэтому x^{n+1} можем вычислять также и по формуле

$$x^{n+1} = x^n + \lambda_n z^n. \tag{0.28}$$

Методы (0.27), (0.28) относятся к методам возможных направлений.

1. О способах выбора шаговых множителей в задачах безусловной оптимизации

Если значение λ_n выбирать из условия

$$f(x^n + \lambda_n z^n) < f(x^n), \tag{0.29}$$

то сходимость (0.21) и даже сходимость (0.23) может отсутствовать. Например, для минимизируемой функции $f(x) \equiv (x+1)^2$, после-

довательность $x_n = 1/n$ удовлетворяет условию (0.29) и тем не менее не сходится к решению $x^* = -1$.

Чтобы обеспечить сходимость (0.23), можно в качестве λ_n выбирать решение λ^* следующей вспомогательной задачи одномерной оптимизации:

$$\lambda^* = \arg\min_{\lambda \geqslant 0} f(x^n + \lambda z^n). \quad (0.30)$$

Однако данный способ применяется лишь в редких случаях (например, в случае квадратичной функции f), так как обычно задача (0.30) чрезмерно трудоемкая. Поэтому, либо вместо задачи (0.30) решают задачу минимизации некоторой более простой функции $F(\lambda)$, аппроксимирующей функцию $f(x^n + \lambda z^n)$ (например, в качестве $F(\lambda)$ часто выбирают параболу $a_n\lambda^2 + b_n\lambda + c$, проходящую через точки $f(x^n)$, $f(x^n + \lambda_{n-1} z^n)$ и $f(x^n + 2^p\lambda_{n-1} z^n)$, если $f(x^n + \lambda_{n-1}z^n) < f(x^n)$, то $p = 1$, иначе $p = -1$), либо вместо условий (0.29) проверяют некоторые более сильные условия, например, условия вида [28]:

$$f(x^n + \lambda_n z^n) < f(x^n) - s\lambda_n^2. \quad (0.31)$$

В последнем случае используется следующий алгоритм 1: для произвольно выбранных чисел $s > 0$, $s_1 > 2$, $\lambda_1 = 1$ полагаем $\lambda_n = 2^{-k_n} s_1\lambda_{n-1}$, где k_n — минимальное натуральное число, при котором выполняется неравенство (0.31).

Лемма 1. Если λ_n выбирать таким образом, чтобы обеспечивалось выполнение неравенства

$$C_n \geqslant \varphi(\alpha_n) \quad (0.32)$$

для некоторой неубывающей положительной при $\alpha_n > 0$ функции φ (где C_n — число из неравенства (0.24)), то при $\inf_x f(x) > -\infty$ в случае $\alpha_n = \| \nabla f(x^n) \|$ будет обеспечена сходимость (0.23), в случае $\alpha_n = \| x^n - x^* \|$ — сходимость (0.22), а в случае $\alpha_n = f(x^n) - \inf_x f(x)$ — будет обеспечена сходимость (0.21).

Действительно, если предположить, что $\lim_{n\to\infty} \alpha_n \neq 0$, то из неравенств (0.24), (0.32) получаем неравенство

$$\lim_{n\to\infty} f(x^{n+1}) = \lim_{n\to\infty} (f(x^{n+1}) - f(x^n) + f(x^n) - f(x^{n-1}) +$$
$$+ \cdots + f(x^2) - f(x^1) + f(x^1)) < \lim_{n\to\infty} \sum_{i=1}^{n} (-\varphi(\alpha_i)) + f(x^1) = -\infty,$$

которое противоречит неравенству $\inf_x f(x) > -\infty$.

Следствие 1. Если для функции f существует такая неубывающая функция $\varphi(\alpha) > 0$ при $\alpha > 0$, что $\| \nabla f(x) \| \geqslant \varphi(f(x) - \inf_x f(x))$, то сходимость (0.23) влечет сходимость (0.21), если $\| \nabla f(x) \| \geqslant \varphi(\| x - x^* \|)$, то сходимость (0.23) влечет сходимость (0.22).

Теорема 1. Если $\nabla f(x)$ *удовлетворяет на множестве* $\{x \mid f(x) \leqslant f(x^1)\}$ *условию Липшица с константой* L*, то для алгоритма* 1 *с* (0.27) *существует функция* φ*, удовлетворяющая условию* (0.32) *при* $\alpha_n = \| \nabla f(x^n) \|$. *Такой функцией является* $\varphi(\alpha) = \min \left\{ s_1\lambda_{n-1}/2, \frac{\alpha}{2(L+s)} \right\}$.

Таким образом, в условиях теоремы 1 при $\inf_x f(x) > -\infty$ алгоритм 1 обеспечивает сходимость (0.23). Методы вычисления λ_n, в которых используются значения $f(x^{n+1})$, $f(x^n)$, ..., называют *адаптивными*. Поэтому все упомянутые выше методы являются адаптивными методами. В неадаптивных методах всю последовательность λ_1, λ_2, ... задают до начала вычисления последовательности x^1, x^2, ... и она не зависит от будущих значений $f(x^1)$, $f(x^2)$, ... При этом, естественно, процесс вычислений x^1, x^2, ... упрощается. Однако здесь не учитываются особенности конкретной минимизируемой функции f и в итоге такой способ выбора λ_n может оказаться менее эффективным, чем в адаптивных алгоритмах. Один класс неадаптивных способов выбора λ_n, обеспечивающих сходимость (0.23) для метода (0.26), дает теорема 2 и ее следствие.

Теорема 2. Если градиент $\nabla f(x)$ *удовлетворяет условию Липшица с константой* L и $\inf_x f(x) > -\infty$, *то метод* (0.28) *удовлетворяет неравенству*

$$\min_{i=\overline{1,n}} \| \nabla f(x^i) \| \leqslant (f(x^1) - \inf_x f(x) - 2^{-1} L \sum_{j=1}^{n} \lambda_j^2)/\delta \sum_{j=1}^{n} \lambda_j. \quad (0.33)$$

Неравенство (0.33) можно получить из неравенств $f(x^{n+1}) \leqslant f(x^n) - \lambda_n \delta \| \nabla f(x^n) \| + 2^{-1} L\lambda_n^2$ справедливых для всех n.

Следствие 1. Если выбранная последовательность λ_1, λ_2, ... удовлетворяет условиям

$$\lambda_j > 0, \quad \sum_{j=1}^{\infty} \lambda_j = \infty, \quad \sum_{j=1}^{\infty} \lambda_j^2 / \sum_{j=1}^{\infty} \lambda_j = 0, \quad (0.34)$$

то в предположениях теоремы 2 $\underline{\lim}_{n\to\infty} \| \nabla f(x^n) \| = 0$.

Действительно, при условиях (0.34) из неравенства $\inf_x f(x) > -\infty$ и неравенства (0.33) получаем $\underline{\lim}_{n\to\infty} \| \nabla f(x^n) \| = 0$.

Для выпуклой функции f выбор λ_n упрощается [153, 404]:

Теорема 3. Если f *— выпуклая функция с минимальным значением* $f(x^*)$, *то при выполнении условий*

$$\lambda_n > 0, \quad \lambda_n \to 0, \quad \sum \lambda_n = \infty, \quad z^n \to \nabla f(x^n) \quad (0.35)$$

для метода (0.28) *выполняется равенство* $\underline{\lim}_{n\to\infty} f(x^n) = f(x^*)$.

Действительно, из очевидного равенства $x^{n+1} - x^0 = x^n - x^0 + \lambda_n z^n$ получаем неравенство $\| x^{n+1} - x^* \|^2 \leqslant \| x^1 - x^* \|^2 -$

$$-\sum_{i=1}^{n}((\nabla f(x^i),\ x^i - x^*) - (\nabla f(x^i) - z^i,\ x^i - x^*) - \lambda_i),$$ которое при условиях (0.35) влечет равенство $\lim_{i\to\infty}(\nabla f(x^i),\ x^i - x^*) = 0$. Поэтому с учетом неравенства $f(x^i) - f(x^*) \leqslant (\nabla f(x^i),\ x^i - x^*)$ (справедливого для выпуклой функции f) получаем $\lim_{i\to\infty} f(x^i) = f(x^*)$.

Очевидно условие $\lambda_n \to 0$ практически невозможно реализовать на ЭВМ с конечной разрядной сеткой. Поэтому на практике часто используют методы как с постоянным шаговым множителем $\lambda_n = \lambda$, так и некоторые промежуточные между адаптивными и неадаптивными [29, 32].

Теорема 4. *Для метода* (0.28) *при* $\lambda_n \equiv \lambda$ *в условиях теоремы 2 выполняется неравенство*

$$\min_{i=\overline{1,n}} \|\nabla f(x^i)\| \leqslant (f(x^1) - \inf_x f(x))/\delta\lambda n + L\lambda/2\delta. \qquad (0.36)$$

Действительно, неравенство (0.36) следует из неравенства (0.33) при $\lambda_n = \lambda$.

Следствие 1. За N, $N \leqslant (f(x^1) - \inf_x f(x))/\lambda^3$, итераций по методу (0.28) при $\lambda_n = \lambda$ найдем точку x^i $(i \leqslant N)$, удовлетворяющую неравенству $\|\nabla f(x^i)\| \leqslant \lambda(\lambda + L/2)/\delta$.

Поэтому постоянное значение λ можно сохранять до момента «стабилизации» значения $\min_{i=\overline{n-k,n}} \|\nabla f(x^i)\|$ (или значения $\min_{i=\overline{n-k,n}} f(x^i)$). Затем λ уменьшаем и опять сохраняем его постоянным до следующего момента стабилизации.

2. О методах второго порядка

Чем сильнее отличается функция f (при удалении x от x^n) от линейной функции F_1, тем меньше эффективность градиентного метода (0.26), так как в таких случаях неравенству (0.24) обычно можно удовлетворить лишь при слишком малых (близких к нулю) значениях C_n. Это вынуждает усложнять вычислительный процесс, привлекая более точные и более сложные аппроксимации для функции f. Например, если функция f достаточно хорошо аппроксимируется в окрестности x^n функцией F_2,

$$F_2(x) \equiv f(x^n) + (\nabla f(x^n),\ x - x^n) + \frac{1}{2}(x - x^n)^T \nabla^2_{xx} f(x^n)(x - x^n),$$

то в качестве x^{n+1} целесообразно выбрать точку $\bar{x}^n$, доставляющую наименьшее значение функции F_2. Так как в случае невырожденного положительного гессиана $\nabla^2_{xx} f(x^n)$ искомая точка $\bar{x}^n$ должна являться решением системы

$$\nabla F_2(x^n) \equiv \nabla f(x^n) + \nabla^2_{xx} f(x^n)(\bar{x}^n - x^n) = 0, \qquad (0.37)$$

то получаем

$$x^{n+1} = \bar{x}^n = -[\nabla^2_{xx} f(x^n)]^{-1} \nabla f(x^n). \qquad (0.38)$$

Такой метод называют *методом Ньютона — Канторовича*. Если в окрестности решения x^* гессиан $\nabla^2_{xx} f(x)$ невырожден и начальное приближение x^1 выбрано достаточно близко к x^*, то метод (0.38) имеет квадратичную скорость сходимости в отличие от геометрической скорости сходимости градиентных методов. Однако из-за необходимости решать систему (0.37), трудоемкость вычисления x^{n+1} по методу (0.38) значительно превышает трудоемкость вычисления x^{n+1} по методу (0.26). Другая трудность реализации метода (0.38) связана с необходимостью выбирать начальное приближение x^1 достаточно близкое к x^*, чтобы точка x^{n+1} не оказалась слишком далеко от x^n, где вследствие нарушения близости функции f и ее локальной квадратичной аппроксимации F_2 может оказаться $f(x^{n+1}) > > f(x^n)$ вместо требуемого $f(x^{n+1}) < f(x^n)$. Чтобы обеспечить сходимость к решению при любом выборе x^1, предлагается вычислять x^{n+1} по формуле

$$x^{n+1} = x^n + \lambda_n (\bar{x}^n - x^n) \qquad (0.39)$$

(λ_n можно вычислять, например, с помощью алгоритма 1).

Чтобы обойти трудности, связанные с решением системы (0.37), предложено ряд модификаций метода (0.38). В одних случаях предлагается вычислять гессиан не во всех точках, а лишь в некоторых точках x^{n_i} и оставлять его без изменений на промежуточных итерациях при $n_i \leq n < n_{i+1}$.

В других случаях предлагается гессиан вообще не вычислять и рассчитывать x^{n+1} по формулам

$$x^{n+1} = x^n - \lambda_n B_n \nabla f(x^n), \qquad (0.40)$$

где $\{B_n\}$ — некоторая последовательность матриц, удовлетворяющая условию $B_n \to \nabla^2_{xx} f(x^n)$. Найдены довольно эффективные формулы для пересчета λ_n, B_n, при которых метод (0.40) сохраняет хорошее свойство метода (0.38) (т. е. имеет квадратичную скорость сходимости) и в то же время является менее трудоемким. Такие методы названы *квазиньютоновскими*, или *методами с переменной метрикой* (ввиду того, что операторы B_n можно интерпретировать как преобразование метрики в пространстве градиентов).

Различные алгоритмы реализации квазиньютоновских методов разработаны также и для тех практически важных задач оптимизации, когда в силу отсутствия аналитического выражения для функции f (или по другим причинам) градиенты $\nabla f(x^n)$ не вычисляются и в расчетных формулах используются лишь вычисленные в предыдущих точках значения $f(x^n)$, $f(x^{n-1})$, ... В основе таких методов лежат те или иные способы аппроксимации неизвестных градиентов и гессианов с помощью конечных разностей.

Эффективные способы построения матриц B_n предложены в методах оптимизации с растяжением пространства и в близком к ним

методе эллипсоидов. В этих методах на n-й итерации осуществляют «растяжение» пространства в некотором направлении z^n, совпадающим с $\nabla f(x^n)$ или с разностью двух последовательных градиентов, т. е.

$$B_{n+1} = B_n R_{1/\alpha_{n+1}}(z^n),$$

где $R_{1/\alpha_{n+1}}(z^n)$ — матрица растяжения пространства в направлении z^n с коэффициентом растяжения α_{n+1} (элементы r_{ij} матрицы $R_\alpha(z^n)$ определяются равенствами [404]

$$r_{ij} = (R_\alpha(z^n) e_i, e_j) = \begin{cases} (\alpha - 1) z_i^n z_j^n, & i \neq j; \\ (\alpha - 1) z_i^2 + 1, & i = j. \end{cases}$$

Если f имеет овражную структуру, то z^n чаще ориентировано «поперек» чем «вдоль» оврага и поэтому данное последовательное растяжение пространства уменьшает овражную структуру функции f и тем ускоряет сходимость градиентного метода. Рассмотренный выше метод эллипсоидов, очевидно, является вариантом метода растяжения пространства со специальным выбором шагового множителя λ_n.

3. Методы сопряженных градиентов

Векторы $p^1, p^2, \ldots, p^N$ называют *сопряженными* относительно симметричной, положительно определенной матрицы A, если $(Ap^i, p^j) = 0$ при $i \neq j$ и $(Ap^i, p^i) \neq 0$.

Метод сопряженных градиентов состоит в последовательной минимизации функции $f(x) = \frac{1}{2}(Ax, x) + (g, x)$ по направлениям p^i, т. е. начиная с произвольного $x^1 \in \boldsymbol{R}^N$, вычисляем

$$x^{n+1} = x^n + \lambda_n p^n, \quad \lambda_n = \arg\min_\lambda f(x^n + \lambda p^n). \qquad (0.41)$$

Утверждение 14. Метод сопряженных градиентов минимизирует функцию $f(x) = \frac{1}{2}(Ax, x) + (g, x)$ за N итераций, т. е.

$$x^N = x^* = \arg\min_x f(x).$$

Действительно, если ввести новое скалярное произведение $(x, y)_1 \triangleq (Ax, y)$, то метод сопряженных градиентов оказывается эквивалентным минимизации $f(x) = \frac{1}{2}(x, x)_1 + (s, x)_1$ по координатным направлениям.

Распространение этой идеи на случай нелинейной функции f приводит к расчетным формулам (0.41) при

$$p^{1+kN} = -\nabla f(x^{1+kN}),$$
$$p^{n+1} = \|\nabla f(x^{n+1})\|^2 p^n / \|\nabla f(x^n)\|^2 - \nabla f(x^{n+1}) \qquad (0.42)$$

(или при

$$p^1 = -\nabla f(x^1), \quad B_1 = I, \quad p^{n+1} = -B_{n+1} \nabla f(x^n),$$

$$q^n = \nabla f(x^{n+1}) - \nabla f(x^n), \quad s^n = x^{n+1} - x^n, \tag{0.43}$$

$$B_{n+1} = B_n - \frac{B_n q^n q^{n*} B_n^*}{q^{n*} B_n q^n} + \frac{s^n s^{n*}}{s^{n*} q^n}\Bigg).$$

Утверждение 15 [470]. Метод (0.41), (0.42) и метод (0.41), (0.43) обеспечивают квадратичную скорость сходимости к единственной точке минимума x^* сильно выпуклой функции f с липшицевым гессианом.

Если λ_n вычислять приближенно, то p^{n+1} рекомендуется вычислять по формулам

$$p^{n+1} = \left[\frac{(\nabla f(x^{n+1}), p^n)}{\|\nabla f(x^n)\|^2} - 1\right] \nabla f(x^{n+1}) + \frac{\|\nabla f(x^{n+1})\|^2}{\|\nabla f(x^n)\|^2} p^n.$$

Матрицы B_n можно вычислять также по более общим формулам:

$$B_{n+1} = B_n - \frac{B_n q^n q^{n*} B_n^*}{q^{n*} B_n q^n} + \frac{s^n s^{n*}}{s^{n*} q^n} + C v^n v^{n*},$$

$$v^n = s^n / q^{n*} s^n - B_n q^n / q^{n*} B_n q^n, \quad C \in \boldsymbol{R}.$$

Данные методы можно интерпретировать как варианты квазиньютоновских методов.

4. О методах штрафов в задачах с ограничениями

Задачу минимизации функции f_0 при дополнительных ограничениях

$$x \in X \triangleq \{x \mid f_i(x) = 0, \ i \in \mathcal{I}^0, \ f_i(x) \leqslant 0, \ i \in \mathcal{I}^-\}$$

можно провести с помощью *метода штрафов* к задаче безусловной оптимизации, которую затем можно решать упомянутыми выше методами безусловной оптимизации. Для этого выбираем некоторую вспомогательную функцию $h(x)$ *(функцию штрафа)*, которая равна нулю, если $x \in X$ и увеличивается с удалением x от X (например,

$$h(x) = C\Big(\sum_{i \in \mathcal{I}^0} f_i^2(x) + \sum_{i \in \mathcal{I}^-} [f_i^+(x)]^2\Big),$$

$C > 0$, $f_i^+(x) = f_i(x)$, если $f_i(x) > 0$ и $f_i^+(x) = 0$, если $f_i(x) \leqslant 0$) и затем, минимизируя вспомогательную функцию $f(x) = f_0(x) + h(x)$, находим $x^*(C) = \arg\min_x f(x)$. Нетрудно убедится в том, что при весьма общих предположениях

$$x^*(C) \xrightarrow[C \to \infty]{} x^* = \arg\min_{x \in X} f_0(x).$$

Если в качестве $h(x)$ выбрать недифференцируемую функцию вида

$$h_1(x) = C\Big(\sum_{i \in \mathcal{I}^0} |f_i(x)| + \sum_{i \in \mathcal{I}^-} f_i^+(x)\Big),$$

то во многих случаях получаем $x^*(C) = x^*$ при сравнительно небольших значениях C (и это весьма существенно, так как увеличение C обычно увеличивает овражную структуру функции f и тем уменьшает эффективность ее минимизации упомянутыми градиентными методами).

Утверждение 16. Если функции f_0, f_i выпуклые и $C \geqslant \max u_i^*$, где u_i^* — множители Лагранжа для функции $f_0(x) + \sum_{i \in \mathcal{I}^-} u_i f_i(x)$, то для h_1 имеем

$$x^*(C) = x^* = \arg \min_{f_i(x) \leqslant 0, i \in \mathcal{I}^-} f(x).$$

Действительно, если функции f_0, f_i, $i \in \mathcal{I}^-$ — выпуклые, то (x^*, u^*) — седловая точка для функции Лагранжа $f_0(x) + \sum_{i \in \mathcal{I}^-} u_i f_i(x)$, т. е. имеем неравенства

$$f_0(x^*) \leqslant f_0(x) + \sum_{i \in \mathcal{I}^-} u_i^* f_i(x) \leqslant f_0(x) + \sum_{i \in \mathcal{I}^-} u_i^* f_i^+(x) \leqslant h_1(x) + f_0(x),$$

которые вместе с равенством $h_1(x^*) = 0$ дают $x^*(C) = x^*$.

Однако функция $h_1(x)$ недифференцируема и для ее минимизации требуются специальные методы (см. минимизация негладких функций).

Ряд методов с гладкими штрафами описан в гл. V.

5. О методах последовательных приближений для задач условной оптимизации и минимизация негладких функций

Как отмечалось выше в методах последовательных приближений вычисляют $(k+1)$-е приближение x^{k+1} с помощью решения $\bar{x}^{k+1}$ для упрощенной задачи оптимизации

$$\bar{x}^{k+1} = \arg \min_{x \in X^k} F_k(x).$$

Если выбрать $X^k = X$ (или $X_1(x^k, \lambda_k)$), $F_k = F_1(x^k, \cdot)$ и положить

$$x^{k+1} = (1 - \lambda) x^k + \lambda \bar{x}^{k+1}, \quad \lambda = \arg \min_{\lambda} f_0((1 - \lambda) x^k + \lambda \bar{x}^{k+1}),$$

то получим вариант *метода условного градиента.*

Если выбрать $X^k = X_2(x^k, \lambda_k)$ (или $X_1(x^k, \lambda_k)$), $F_k = F_1(x^k, \cdot)$ и положить $x^{k+1} = \arg \min_{x \in X} \| x - \bar{x}^{k+1} \|^2$, то получим вариант *метода проекции градиента.*

Если положить $x^{k+1} = \bar{x}^{k+1}$, $X^k = X_3(x^k, \lambda_k, e_{i_k}) \cap X$, $F_k = f$, то получим *метод покоординатного спуска* ($i_k = k - ENT(k/n)$).

Если в случае $X = \{x \mid f_i(x) \leqslant 0,\ i \in J\}$ выбрать (см. с. 22)

$$X^k = \{x \mid f_i(x^k) + (\nabla f_i(x^k),\ x - x^k) \leqslant 0,$$
$$i \in J_k \triangleq \operatorname{Arg}_i [f_i(x^k) \geqslant C_k - \varepsilon\lambda_k],$$
$$x \in X_2(x^k, \lambda_k)\},\quad C_k \triangleq \max_{i \in \mathcal{I}} f_i(x^k),\quad F_k = F_1(x^k, \cdot),\quad x^{k+1} = \bar{x}^{k+1},$$

то получим вариант *метода линеаризации* и *метода возможных направлений*, выбрав $\alpha_k = \operatorname{sign}(C_k - \varepsilon\lambda_k)^+$

$$x^{k+1} = x^k + \lambda_k [(\alpha_k - 1)\nabla f_0(x^k) - \alpha_k \sum_{i \in \mathcal{I}_k} \nabla f_i(x^k)]$$

получим вариант *квазиградиентного метода.*

При негладкой аппроксимации $F_k(x) = F_3(x^k, x)$ имеем

$$F'_k(x^k, z) = \sum_i a_i(x^k) \max_{j \in \mathcal{I}_1(x^k, i)} (c^{ij}(x^k), z)$$
$$(\mathcal{I}_1(x^k, i) \triangleq \arg \max_{j \in \mathcal{I}(x^k, i)} F_3(x^k, x^k))$$

и поэтому можем вычислять направления убывания f и направления наискорейшего убывания $z(x^k) = \arg \min_{\|z\|=1} F'_k(x^k, z)$. Если вычисление $z(x^k)$ затруднительно (как, например, в стохастическом случае $f(x) \triangleq M_\omega F(x, \omega)$), то удобнее использовать метод $x^{k+1} = x^k - \lambda_k \xi_k / \|\xi_k\|$, $\xi_k = \xi(F_3, x^k) \triangleq \sum_i a_i(x^k) c^i_{j_{ki}}$, $j_{ki} \in \mathcal{I}_1(x^k, i)$ (в стохастическом случае [153] $\xi_k = b_k(\omega^k)$, $M_\omega b_k(\omega) \to \xi(f, x^k)$), который для выпуклой задачи обеспечивает $f(x^k) \to f(x^*)$ при $\lambda_k > 0$, $\sum_{k=1}^{\infty} \lambda_k = \infty$, $\lambda_k \to 0$, $\left(\sum_{k=1}^{\infty} \lambda_k^2 < \infty\right)$, для слабовыпуклой (при дополнительном условии $\lambda_{k+1}/\lambda_k \to 1$) имеем $f(x^k) \to f(x^*)$ (в стохастическом случае полагаем $x^{k+1} = x^0$, если $\|x^{k+1}(\omega)\| > C$ [272]), а для f, касательных к F_3, обеспечивает $\|z(x^k)\| < \varepsilon$ [32]. В [32] получены условия сходимости для конечных $\lambda_k = \lambda$, поскольку на ЭВМ трудно обеспечивать $\lambda_k \to 0$.

Методы развязывающей декомпозиции для общей задачи

$$x^* = \arg\min_{x \in X} f_0(x),$$
$$X \triangleq \{x = (x_1, x_2) \mid f_i(x) = 0,\ i \in J_1,\ f_i(x) \leqslant 0,\ i \in J_2,\ x_2 \in X_2\} \qquad (0.44)$$

основаны на следующем утверждении [23].

Утверждение 17. Если для всех $x_2 \in X_2$ и всех i из множества $\mathcal{I}_3$ существенных ограничений $\mathcal{I}_3 \subset J_1 \cup J_2 \setminus \{0\}$ вспомогательные функции f_i^1, B_i, g и B_0 удовлетворяют условиям

$$f_i(x) = f_i^1(x_1, x_2, g(x)),$$
$$\varphi_0(x) \triangleq g(x) - B^0(x_2, f_{\mathcal{I}_3}(x)) \equiv 0,$$
$$\varphi_i(x) \triangleq f_i(x) - B_i(x_2, g(x), f_{\mathcal{I}_3}(x)) \equiv 0, \qquad (0.45)$$

то оптимальное значение $\overset{*}{x_2}$ равно

$$\arg\min_{x_2 \in X_2} B_0(x_2,\ B^0(x_2,\ 0),\ 0) \mid B_i(x_2,\ B^0(x_2,\ 0),\ 0) = 0, \quad i \in \mathcal{I}_3. \quad (0.46)$$

Задача (0. 46)обычно имеет существенно меньшую размерность, чем задача (0.44). Несколько бо́льшую размерность имеет эквивалентная задача минимизации штрафной функции

$$f_0(x) + C_1 \| g(x) - u \| + \sum_{i \in J_2 \setminus \mathcal{I}_3} C_{2i} f_i^+(x),$$

или функции

$$f_0(x) + \tilde{F}(x),$$

$$\tilde{F}(x) \triangleq \max \{0,\ \max_j |u_j - g_j(x)|,\ \max_{i \in J_2 \setminus \mathcal{I}_3} f_i(x)\},$$

при связях [22, 24]

$$f^1_{\mathcal{I}_3}(x_1,\ x_2,\ u) = 0.$$

Функцию B_i называют развязывающим оператором для функции f_i по переменной x_1 на множестве X_2 относительно функции $f_{\mathcal{I}_3}$. Ее аппроксимация B_i^k, обеспечивающая в точке x^k нулевые значения всем производным до s-го порядка по всем направлениям в пространстве переменной x_1 для функции φ_i, определяет в точке x^k асимптотически развязывающий оператор s-го порядка

$$B_i^s(x_2) \triangleq f_i(\bar{x}_1,\ x_2) + B_i^k(x_2,\ B^0(x_2,\ 0),\ 0) -$$
$$- B_i^k(x_2,\ B^0(x_2,\ f_{\mathcal{I}_3}(\bar{x}_1,\ x_2)),\ f_{\mathcal{I}_3}(\bar{x}_1,\ x_2)),$$

обладающий следующим важным свойством [23].

Утверждение 18. $f_{\mathcal{I}_3}(x) = 0 \Rightarrow f_i(x) = B_i^s(x_2) + o(\| x_1 - \bar{x}_1 \|^s)$.

Поэтому естественно выбирать в качестве указанных выше X^k и F_k соответственно X_2 и

$$B^s(x_2) \triangleq (1 - \alpha_k) B_0^s(x_2) + \sum_{i \notin \mathcal{I}_3} \max \{0,\ \mathrm{sign}\, B_i^s(x_2)\} B_i^s(x_2)$$

и вычислять $\bar{x}^{k+1}$ с помощью известных методов недифференцируемой оптимизации (малая размерность вектора x_2 позволяет здесь эффективно воспользоваться методами растяжения пространства). Приведем общий A_3-алгоритм. Начало: выбираем произвольные $x_2^1 \in X_2$, положительные числа λ_1, C_1, C_2 и C_3 и полагаем $k = 1$. Основной цикл: вычисляем $\bar{x}_2^{k+1}$ для

$$X^k = \{x_2 \in X_2 \mid \| x_2 - x_2^k \| \in [\lambda_k,\ C_2\lambda_k]\},$$

где λ_k — максимальное число из последовательности $\{C_3\lambda_{k-1}\, 2^{-m}\}_{m=0}^{\infty}$, удовлетворяющее неравенству

$$f_0(\bar{x}^{k+1}) \leqslant f_0(x^k) + C_1\lambda_k^{s+1}.$$

Утверждение 19. Если $f_0(x^*) > -\infty$ и f_i являются функционалами равномерного роста s-го порядка на X_2 (например, ограничены все производные f_i до $(s+1)$-го порядка по всем допустимым направлениям), то подпоследовательность $\{x_2^{k_i}\}$ сходится за конечное число итераций $k = k(\varepsilon)$ к ε-экстремальному решению u_ε^* s-го порядка (т. е. производная s_1-го порядка функции f_0 в точке u_ε^* по любому допустимому направлению не меньше $-\varepsilon^{s+1+s_1}$ для всех $s_1 = \overline{0, s}$).

Реализуемые Bδ-адекватные модели. Выше уже отмечалось, что при решении практических задач функции f_i обычно являются всего лишь некоторыми приближениями к соответствующим либо неизвестным, либо слишком сложным для современных ЭВМ «реальным» функциям $\bar{f}_i$ (т. е. исходная математическая модель обычно либо неизвестна, либо нереализуема). Поэтому актуальной является проблема выбора таких приближений f_i (т. е. формулировки такой задачи оптимизации), чтобы, во-первых, ее приближенное решение x_f^* не слишком отличалось от искомого $x_{\bar{f}}^*$ и, во-вторых, существовала бы практическая возможность вычислить x_f^* в требуемые сроки на имеющейся ЭВМ. Одним из инструментов для выбора и адаптивного уточнения таких функций является следующая теорема.

Теорема о реализуемой Bδ-адекватной модели. *Если неизвестные (или нереализуемые) функции $\bar{f}_0, \bar{f}_i$ в реальной задаче оптимизации аппроксимированы реализуемыми функциями f_0, f_i, а развязывающие операторы B_i аппроксимированы функциями $\bar{B}_i$, удовлетворяющими на приближении $\bar{x}$ к решению x_f^* условиям*

$$|f_i(\bar{x}_1, x_2) - \bar{f}_i(\bar{x}_1, x_2)| \leqslant \delta_{1i}(f_0, \bar{x}_1) \triangleq \delta_{1i}(f, \bar{x}_1),$$

$$|\bar{B}_i(x_2, \bar{f}_{\mathcal{J}_3}(\bar{x}_1, x_2)) - B_i(x_2, f_{\mathcal{J}_3}(\bar{x}_1, x_2))| \leqslant \delta_{2i}(f, \bar{x}_1),$$

$$|\bar{f}_i(x) + \bar{B}_i(x_2, \bar{f}_{\mathcal{J}_3}(x)) - \bar{f}_i(\bar{x}_1, x_2) - \bar{B}_i(x_2, \bar{f}_{\mathcal{J}_3}(\bar{x}_1, x_2))| \leqslant$$
$$\leqslant \delta_{3i}(f, \bar{x}_1)$$

и $\sum_{j=1}^{3} \delta_{ji}(f, \bar{x}_1) \leqslant \delta_i$, *то функция*

$$\hat{B}_i(x_2) \triangleq f_i(\bar{x}_1, x_2) + \bar{B}_i(x_2, f_{\mathcal{J}_3}(\bar{x}_1, x_2)) - \bar{B}_i(x_2, 0)$$

определяет реализуемую Bδ-адекватную модель для $\bar{f}_i$, т. е. удовлетворяет неравенству

$$|\bar{f}_i(x) - \hat{B}_i(x_2)| \leqslant \delta_i.$$

Эта теорема указывает пути целесообразного перераспределения вычислительного ресурса ЭВМ как на уточнение $\bar{x}$, так и на уточнение f с целью быстрейшего уменьшения и $f_0(\bar{x})$ и δ_0 $(\delta_{\mathcal{J}})$. Для этого

очевидно достаточно находить зависимость скорости убывания величин δ_{ij} от объемов вычислений при уточнении $\bar{x}$, f и $\bar{B}$ и дальше перераспределять ресурсы на уточнение либо $\bar{x}$, либо f или $\bar{B}$ в зависимости от того, где обеспечивается наиболее быстрое убывание f_0, δ_0, а значит и $\bar{f}_0$. Практическая эффективность таких алгоритмов зависит от удачного выбора множеств $\mathcal{I}_3$, так как этим определяется и объем вычислений при разрешении системы $f^1_{\mathcal{I}_3}(x_1, x_2, u) = 0$ и скорость приближения x^k к x^*. С этой целью часто выбирают множества

$$\mathcal{I}_3 \triangleq \mathcal{I}_3^k \triangleq \mathcal{I}_3(x^k, \varepsilon_k) \triangleq \{i \mid f_i(x^k) \geqslant \tilde{F}(x^k) - \varepsilon_k\} \cup$$
$$\cup \{j \mid |u_j - g_j(x^k)| \geqslant \tilde{F}(x^k) - \varepsilon_k\}$$

с небольшими значениями $\varepsilon_k > 0$ и строят аппроксимации обратной функции $(f^1_{\mathcal{I}_3})^{-1}$ (или обратного гессиана) и развязывающих операторов в некотором выбранном параметрическом классе функций вида $\hat{f}(p_k, \mathcal{I}_3^k, x, u, f^1_{\mathcal{I}_3}(x, u))$. Параметры p_k оценивают и уточняют по вычисленным значениям $x^1, x^2, \ldots, x^k$.

В случае аппроксимации обратного гессиана удобно использовать матричное тождество

$$\begin{pmatrix} A_1 + KL & C \\ D & A_2 \end{pmatrix}^{-1} = \begin{pmatrix} A_3 + A_3CMDA_3 & -A_3CM \\ -MDA_3 & M \end{pmatrix}, \qquad (0.47)$$

где

$$A_3 = A_1^{-1}(E - K(E + LA_1^{-1}K)^{-1}LA_1^{-1}) = (A_1 + KL)^{-1},$$
$$M = (A_2 - DA_3C)^{-1},$$

которое получаем как следствие утверждения 17 для линейных f.

С помощью (0.47) получаем эффективный A_6-алгоритм для вычисления (псевдо)обратной матрицы A^{-1} к матрице $A = \{a_{ij}, i = \overline{1, n}\}$: $Z^0 = E$, $C > 0$,

$$z_i^{k+1} = z_i^k + z_{j_k}^k(\delta_{ij_k} - (a^{j_k}, z_i^k))/(a^{j_k}, z_{j_k}^k),$$

где

$$j_k = \arg\min_j \varphi(j), \quad \text{или} \quad j_k = \arg_j[\varphi(j) \leqslant C], \qquad (0.48)$$
$$\varphi(j_k) = \|Z^{k+1}\| \quad \text{или} \quad \varphi(j_k) = |(a^{j_k}, z_{j_k}^k)|^{-1}.$$

Утверждение 20. Если на некотором шаге $k < n$ задача (0.48) не разрешима, то Z^k является C-псевдообратной к A, иначе $Z^n = A^{-1}$, а при

$$v = \arg_v[(E + (A^1 - E)Z^k)v = (A^1 - E)z^k,$$
$$a_{ij}^1 = a_{ij}, \quad v_i a_{ij}^1 = 0]$$

имеем

$$A^{-1}b = Z^k b - Z^k v.$$

Ошибки округлений в Z^n можно существенно уменьшить с помощью A_7-алгоритма:

$$B_0 = Z^n, \quad b_0 = b,$$

$$B_{k+1} = B_k + (E - B_i A)^{2^i} B_k, \quad x^{k+1} = x^k + (E - B_j A)^{2^j} x^k,$$

обеспечивающего при i, $j \leqslant n$ быструю сходимость $B_k \to A^{-1}$, $x^k \to$ $\to A^{-1}b$ согласно следующим утверждениям [27].

Утверждение 21. Если $b_{n+1} = b_n + B^{2^n} b_n$, $b_0 = b$, $BA = -C$, $Ax = b - Cx$, то

$$Ax = b_{n+1} - B^{2^{n+1}-1} Cx.$$

Следствие. Если $B_0 = Z^n$, $C = E - B_0 A$, $b_0 = b$,

$$B_{k+1} = B_k + C^{2^k} B_k, \quad x^{k+1} = x^k + C^{2^k} x^k,$$

то

$$B_{k+1} = A^{-1} + C^{2^{k+1}} A^{-1}, \quad x^{k+1} = A^{-1}b + C^{2^{k+1}} A^{-1} b.$$

Подбором i, j в A_7-алгоритме уменьшается влияние ошибок округлений.

Часть I

МЕТОДЫ ОДНОМЕРНОЙ И БЕЗУСЛОВНОЙ ОПТИМИЗАЦИИ

Глава 1

МЕТОДЫ ОДНОМЕРНОЙ ОПТИМИЗАЦИИ

1.1. Методы Фибоначчи

Задача 0. Найти $\arg \min\limits_{x \in [a_0, b_0]} f_0(x)$ для заданной функции f_0: $R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 0. Функция f_0 такова, что на отрезке $[a_0, b_0]$ точка ее локального минимума x^* является точкой абсолютного минимума f_0 на отрезке $[a_0, b_0]$.

Методы Фибоначчи являются оптимальными (для класса функций f_0, удовлетворяющих предположению 0) по количеству вычислений минимизируемой функции f_0 при заданной точности вычисления x^*.

1. Основной алгоритм

Алгоритм 1

Начало. I. Выбрать число $\varepsilon > 0$ — точность вычисления точки минимума функции f_0 на отрезке $[a_0, b_0]$; положить $F_1 = F_2 = 1$.

II. Положить $j = 1$.

III. Вычислить число $F_{j+2} = F_{j+1} + F_j$.

IV. Если $F_{j+1} < \frac{1}{\varepsilon}(b_0 - a_0) \leqslant F_{j+2}$, то положить $m = j$ и перейти к шагу V; иначе положить $j = j + 1$ и перейти к шагу III.

V. Вычислить точки

$$y_1 = a_0 + \frac{F_m}{F_{m+2}}(b_0 - a_0);$$

$$z_1 = a_0 + b_0 - y_1.$$

VI. Если $f_0(y_1) \leqslant f_0(z_1)$, то положить $a_1 = a_0$, $b_1 = z_1$ и перейти к шагу VII; иначе положить $a_1 = y_1$, $b_1 = b_0$ и перейти к шагу VII.

VII. Положить $k = 1$.

Основной цикл. VIII. Если $f_0(y_k) \leqslant f_0(z_k)$, то вычислить точку $y_{k+1} = a_k + b_k - y_k$, вычислить значение $f_0(y_{k+1})$ и перейти к шагу IX; иначе положить $y_{k+1} = z_k$, $f_0(y_{k+1}) = f_0(z_k)$ и перейти к шагу X.

IX. Положить $z_{k+1} = y_k$; $f_0(z_{k+1}) = f_0(y_k)$ и перейти к шагу XI.

X. Вычислить точку $z_{k+1} = a_k + b_k - z_k$, вычислить значение $f_0(z_{k+1})$ и перейти к шагу XI.

XI. Если $f_0(y_{k+1}) \leqslant f_0(z_{k+1})$, то положить $a_{k+1} = a_k$, $b_{k+1} = z_{k+1}$ и перейти к шагу XII; иначе положить $a_{k+1} = y_{k+1}$, $b_{k+1} = b_k$ и перейти к шагу XII.

XII. Если $k < m - 1$, то положить $k = k + 1$ и перейти к шагу VIII; иначе положить $\bar{x}^* = (a_m + b_m)/2$ и прекратить вычисления.

Теорема 1. Если выполнено предположение 0, то для произвольного $\varepsilon > 0$ *точка* $\bar{x}^*$, *порожденная алгоритмом 1, удовлетворяет неравенству* $| x^* - \bar{x}^* | \leqslant \varepsilon$.

Замечание 1. Недостатком алгоритма 1 является то, что погрешности в вычислениях точек y_k, z_k могут столь быстро накапливаться, что ожидаемая точность решения будет существенно отличаться от реальной. Следующая модификация метода Фибоначчи менее чувствительна к погрешностям вычислений.

2. Модификация метода Фибоначчи

Алгоритм 2.

Начало. Шаги I—IV такие, как в алгоритме 1.

V. Вычислить точки

$$y_1 = a_0 + \frac{F_m}{F_{m+2}}(b_0 - a_0);$$

$$z_1 = a_0 + \frac{F_{m+1}}{F_{m+2}}(b_0 - a_0).$$

VI. Если $f_0(y_1) \leqslant f_0(z_1)$, то положить $a_1 = a_0$, $b_1 = z_1$ и перейти к шагу VII; иначе положить $a_1 = y_1$, $b_1 = b_0$ и перейти к шагу VII.

VII. Положить $k = 1$.

Основной цикл. VIII. Если $f_0(y_k) \leqslant f_0(z_k)$, то вычислить точку

$$y_{k+1} = a_k + \frac{F_{m-k}}{F_{m+2}}(b_0 - a_0),$$

вычислить значение $f_0(y_{k+1})$ и перейти к шагу IX; иначе положить $y_{k+1} = z_k$, $f_0(y_{k+1}) = f_0(z_k)$ и перейти к шагу X.

IX. Положить $y_{k+1} = z_k$, $f_0(y_{k+1}) = f_0(z_k)$ и перейти к шагу XI.

X. Вычислить точку $z_{k+1} = a_k + \frac{F_{m-k+1}}{F_{m+2}}(b_0 - a_0)$, вычислить значение $f_0(z_{k+1})$ и перейти к шагу XI.

XI. Если $f_0(y_{k+1}) \leqslant f_0(z_{k+1})$, то положить $a_{k+1} = a_k$, $b_{k+1} = b_k$ и перейти к шагу XII; иначе положить $a_{k+1} = y_{k+1}$, $b_{k+1} = b_k$ и перейти к шагу XII.

XII. Если $k < m - 1$, то положить $k = k + 1$ и перейти к шагу VIII; иначе положить $\bar{x}^* = (a_m + b_m)/2$ и прекратить вычисления.

Точка $\bar{x}^*$ удовлетворяет неравенству $|\bar{x}^* - x^*| \leqslant \varepsilon$, если выполнено предположение 0.

Библиографические указания. При написании параграфа использовались работы [194, 358, 378].

1.2. Метод золотого сечения

Задача 1. Найти $\arg \min\limits_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0 : R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 1. Функция f_0 такова, что на отрезке $[a_0, b_0]$ точка ее локального минимума x^* является точкой абсолютного минимума f_0 на отрезке $[a_0, b_0]$.

Алгоритм 1

Начало. I. Вычислить константу $\alpha = (3 - \sqrt{5})/2$ ($\alpha \cong 0,3819$).

II. Вычислить точки

$$y_1 = a_0 + \alpha(b_0 - a_0);$$
$$z_1 = a_0 + b_0 - y_1$$

и значения $f_0(y_1)$, $f_0(z_1)$.

III. Если $f_0(y_1) \leqslant f_0(z_1)$, то положить $a_1 = a_0$, $b_1 = z_1$ и перейти к шагу IV; иначе положить $a_1 = y_1$, $b_1 = b_0$ и перейти к шагу IV.

IV. Положить $k = 1$.

Основной цикл. V. Если $f_0(y_k) \leqslant f_0(z_k)$, то вычислить $y_{k+1} = a_k + b_k - y_k$, $f_0(y_{k+1})$ и перейти к шагу VI; иначе положить $y_{k+1} = z_k$, $f_0(y_{k+1}) = f_0(z_k)$ и перейти к шагу VII.

VI. Положить

$$z_{k+1} = y_k, \quad f_0(z_{k+1}) = f_0(y_k)$$

и перейти к шагу VIII.

VII. Вычислить $z_{k+1} = a_k + b_k - z_k$, $f_0(z_{k+1})$ и перейти к шагу VIII.

VIII. Если $f_0(y_{k+1}) \leqslant f_0(z_{k+1})$, то положить $a_{k+1} = a_k$, $b_{k+1} = z_{k+1}$ и перейти к шагу IX; иначе положить $a_{k+1} = y_{k+1}$, $b_{k+1} = b_k$ и перейти к шагу IX.

IX. Вычислить $x^k = (a_{k+1} + b_{k+1})/2$.

X. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если выполнено предположение 1, то последовательность $\{x^k\}_{k+1}^{\infty}$, порожденная алгоритмом 1, такова, что*

$$\lim_{k\to\infty} f_0(x^k) = f_0(x^*); \quad \lim_{k\to\infty} |x^k - x^*| = 0.$$

Замечание 1. Длина отрезка $[a_k, b_k]_{\text{з.с}}$, построенного по методу золотого сечения, на 17 % больше длины отрезка $[a_k, b_k]_{\text{Фиб}}$, построенного по методу Фибоначчи. Однако метод золотого сечения обладает тем достоинством, что на каждой его итерации приходится делать меньше вычислений.

Замечание 1'. Иногда на практике комбинируют оба метода: первые шаги делают по методу золотого сечения, а когда оптимум достаточно близок, вычисляют число m и переходят к методу Фибоначчи.

Библиографические указания. При написании параграфа использовались работы [194, 358, 378].

1.3. Оптимальный метод поиска экстремума унимодальных функций, удовлетворяющих условию Липшица

Задача 1. Найти $\arg \min_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0 : \mathbf{R}^1 \to \mathbf{R}^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 1. (i) — функция f_0 определена, непрерывна и удовлетворяет условию Липшица с константой γ на интервале $[a_0, b_0]$; (ii) — функция f_0 унимодальна, т. е. имеет на отрезке $[a_0, b_0]$ единственную точку минимума, слева от которой f_0 строго убывает, а справа строго возрастает с ростом x.

Приведенный ниже алгоритм является оптимальным для класса функций, удовлетворяющих предположениям 1, т. е. дает наименьшее гарантированное значение длины интервала, содержащего минимум функции f_0, после вычисления ее в заданном числе N точек. В k-й итерации алгоритма в начале находят (используя константу γ и значения функции f_0 в точках $x_1, x_2, \ldots, x_{k-1}$) интервал локализации $[a_1, b_1]$ точки минимума функции f_0, затем внутри интервала $[a_1, b_1]$ — точку x_k, в которой следует вычислить функцию f_0. При вычислении интервала локализации учитывается конечность константы Липшица γ, что в практических задачах приводит к значительному выигрышу (по сравнению, например, с методом Фибоначчи, который является оптимальным на классе унимодальных функций при $\gamma = \infty$).

Алгоритм 1

Начало. I. Задать константу N — число вычислений функции f_0.

II. Найти числа Фибоначчи $F_0, F_1, \ldots, F_{N+1}$, определяемые соотношениями

$$F_0 = F_1 = 1, \; F_{n+1} = F_{n-1} + F_n, \; n = 1, 2, \ldots .$$

III. Положить $a_1 = a_0$, $b_1 = b_0$.

IV. Положить $k = 1$.

Основной цикл. V. Если $k = 1$, то перейти к шагу VIII; иначе перейти к шагу VI.

VI. Вычислить

$$f^*_{k-1} = \min_{1 \leqslant i \leqslant k-1} f_0(x_i). \quad (1.1)$$

VII. Если значение f^*_{k-1} в (1.1) достигается в двух соседних точках x_{k_1}, x_{k_2}, $k_1 \in [1 : (k-1)]$, $k_2 \in [1 : (k-1)]$, таких, что $x_{k_1} <$

$< x_{k_2}$, то положить $a_1 = x_{k_1}$, $b_1 = x_{k_2}$ и перейти к шагу VIII; если значение f^*_{k-1} в (1.1) достигается в одной точке x_{k_1}, $k_1 \in [1 : (k-1)]$, то положить $\overset{*}{x}_{k-1} = x_{k_1}$ и перейти к шагу X.

(Последующие шаги VIII—IX алгоритма соответствуют той ситуации, когда внутри интервала $[a_1, b_1]$ не проводились вычисления функции f_0).

VIII. Если $k = N + 1$, то прекратить вычисления (в этом случае точка минимума функции f_0 принадлежит интервалу $[a_1, b_1]$, длина которого не превышает гарантированного значения $(b_0 - a_0)/F_N$; иначе перейти к шагу IX.

IX. Вычислить точку x_k следующего вычисления функции f_0

$$x_k = a_1 + (b_1 - a_1)\, F_{N-k+1}/F_{N-k+2}$$

$$(\text{или } x_k = a_1 + (b_1 - a_1)\, F_{N-k}/F_{N-k+2});$$

вычислить $f_0(x_k)$; положить $k = k + 1$ и перейти к шагу V.

(Шаги X—XVIII алгоритма соответствуют той ситуации, когда внутри интервала локализации $[a_1, b_1]$ точки минимума существует точка $\overset{*}{x}_{k-1}$, в которой на предыдущих итерациях проводилось вычисление функции f_0).

X. Если функция f_0 не вычислялась слева и справа от точки $\overset{*}{x}_{k-1}$ (такой случай имеет место при $k = 2$), то положить $a_1 = a_0$, $b_1 = b_0$ и перейти к шагу XIV; если для всех $i \in [1 : (k-1)]$ выполняется неравенство $\overset{*}{x}_{k-1} \leqslant x_i$ (т. е. если функция f_0 не вычислялась слева от точки $\overset{*}{x}_{k-1}$), то найти точку x_{k_3}, $k_3 \in [1 : (k-1)]$, $x_{k_3} \neq \overset{*}{x}_{k-1}$, ближайшую к точке $\overset{*}{x}_{k-1}$ с правой стороны, положить $\bar{b}_1 = x_{k_3}$ и перейти к шагу XI;

если для всех $i \in [1 : (k-1)]$ выполняется неравенство $\overset{*}{x}_{k-1} \geqslant x_i$ (т. е. если функция f_0 не вычислялась справа от точки $\overset{*}{x}_{k-1}$), найти точку x_{k_4}, $k_4 \in [1 : (k-1)]$, $x_{k_4} \neq \overset{*}{x}_{k-1}$, ближайшую к точке $\overset{*}{x}_{k-1}$ с левой стороны, положить $\bar{a}_1 = x_{k_4}$ и перейти к шагу XII;

если функция f_0 вычислялась слева и справа от точки $\overset{*}{x}_{k-1}$, то найти точку x_{k_5}, $k_5 \in [1 : (k-1)]$, $x_{k_5} \neq \overset{*}{x}_{k-1}$, ближайшую к $\overset{*}{x}_{k-1}$ слева, и точку x_{k_6}, $k_6 \in [1 : (k-1)]$, $x_{k_6} \neq \overset{*}{x}_{k-1}$, ближайшую к $\overset{*}{x}_{k-1}$ справа, положить $\bar{a}_1 = x_{k_5}$, $\bar{b}_1 = x_{k_6}$ и перейти к шагу XIII.

XI. Найти интервал локализации $[a_1, b_1]$ точки минимума функции f_0 (после вычисления функции f_0 в $k-1$ точках $x_1, x_2, \ldots, x_{k-1}$) по правилу

$$a_1 = a_0;$$

$$b_1 = \bar{b}_1 - \frac{1}{\gamma}\left(f_0(\bar{b}_1) - f^*_{k-1}\right)$$

и перейти к шагу XIV.

XII. Найти интервал локализации $[a_1, b_1]$ точки минимума функции f_0 (после вычисления функции f_0 в $k-1$ точках $x_1, x_2, \ldots, x_{k-1}$)

по правилу

$$a_1 = \bar{a}_1 + \frac{1}{\gamma}(f_0(\bar{a}_1) - f^*_{k-1});$$

$$b_1 = b_0$$

и перейти к шагу XIV.

XIII. Найти интервал локализации $[a_1, b_1]$ точки минимума функции f_0 (после вычисления функции f_0 в $k - 1$ точках $x_1, x_2, \ldots, x_{k-1}$) по правилу

$$a_1 = \bar{a}_1 + \frac{1}{\gamma}(f_0(\bar{a}_1) - f^*_{k-1});$$

$$b_1 = \bar{b}_1 - \frac{1}{\gamma}(f_0(\bar{b}_1) - f^*_{k-1})$$

и перейти к шагу XIV.

XIV. Если $k = N + 1$, то прекратить вычисления (в этом случае точка минимума функции f_0 принадлежит интервалу $[a_1, b_1]$, длина которого не превышает гарантированного значения $(b_0 - a_0)/F_N$); иначе перейти к шагу XV.

XV. Вычислить

$$c_k = (x^*_{k-1} - a_1)/(b_1 - a_1).$$

XVI. Вычислить

$$t_k = \begin{cases} 1 - (1 - c_k) F_{N-k+1}/F_{N-k+2}, & \text{если } 0 \leqslant c_k \leqslant 1/2; \\ c_k F_{N-k+1}/F_{N-k+2}, & \text{если } 1/2 \leqslant c_k \leqslant 1. \end{cases}$$

XVII. Вычислить точку

$$x_k = a_1 + (b_1 - a_1) t_k.$$

XVIII. Вычислить значение $f_0(x_k)$, положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Пусть выполняются предположения 1 и $[a_1, b_1]$ — интервал локализации точки минимума функции f_0, вычисленный в k-й итерации ($k = 1, \ldots, N$). Тогда за оставшиеся $N - k + 1$ итераций алгоритм 1 приводит к интервалу локализации точки минимума функции f_0, длина которого не превышает значения $(b_1 - a_1) \times \times s_{N-k+1}$, где*

$$s_{N-k+1} = \begin{cases} \max\{(1 - c_k)/F_{N-k+1}, c_k/F_{N-k}\}, & 0 \leqslant c_k \leqslant 1/2; \\ \max\{(1 - c_k)/F_{N-k}, c_k/F_{N-k+1}\}, & 1/2 \leqslant c_k \leqslant 1, \end{cases}$$

если внутри интервала локализации $[a_1, b_1]$ точки минимума вычисления функции f_0 проводились;

$$s_{N-k+1} = 1/F_{N-k+1},$$

если внутри интервала локализации $[a_1, b_1]$ точки минимума вычисления функции f_0 не проводились.

Замечание 1. Если $\gamma = \infty$, то алгоритм 1 дает оптимальный гарантированный результат, равный $(b_0 - a_0) F^{-1}_{N+1}$, т. е. такой же,

как и метод Фибоначчи. На практике учет конечности значения константы Липшица γ в алгоритме 1 приводит к значительному выигрышу (по сравнению с методом Фибоначчи), который получается за счет специфического построения интервала локализации $[a_1, b_1]$ точки минимума на каждой итерации.

Библиографические указания. Параграф написан на основании работ [386, 389].

1.4. Оптимальный метод поиска экстремума выпуклых функций

Задача 1. Найти $\arg \min\limits_{x \in [a_0, b_0]} f_0(x)$ для заданной функции f_0: $R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 1. Функция f_0 выпукла на интервале $[a_0, b_0]$.

Приведенный ниже алгоритм является оптимальным на классе выпуклых функций, т. е. дает наименьшее гарантированное значение длины интервала, содержащего минимум функции f_0, после вычисления функции f_0 в заданном числе N точек.

В k-й итерации алгоритма по известным значениям функции f_0 в точках $x_1, x_2, \ldots, x_{k-1}$ строится интервал локализации $[x_-, x_+]$ точки минимума функции f_0, затем внутри интервала локализации $[x_-, x_+]$ находят точку x_k, в которой следует вычислить функцию f_0.

Алгоритм 1

Начало. I. Задать константу N — число вычислений функции f_0.

II. Найти числа Фибоначчи $F_0, F_1, \ldots, F_{N+1}$, определяемые соотношениями

$$F_0 = F_1 = 1, \quad F_{n+1} = F_{n-1} + F_n, \quad n = 1, 2, \ldots .$$

III. Положить $k = 1$.

Основной цикл. IV. Если $k = 1$, то положить $x_- = a_0$, $x_+ = b_0$ и перейти к шагу VII; иначе перейти к шагу V.

V. Вычислить

$$f^*_{k-1} = \min_{1 \leq i \leq k-1} f_0(x_i). \tag{1.2}$$

VI. Если значение f^*_{k-1} в (1.2) достигается в двух соседних точках x_{k_1}, x_{k_2}, $k_1 \in [1 : (k-1)]$, $k_2 \in [1 : (k-1)]$, таких, что $x_{k_1} < x_{k_2}$, то положить $x_- = x_{k_1}$, $x_+ = x_{k_2}$ и перейти к шагу VII; если значение f^*_{k-1} в (1.2) достигается в одной точке x_{k_1}, $k_1 \in [1 : (k-1)]$, то положить $\overset{*}{x}_{k-1} = x_{k_1}$ и перейти к шагу X; если значение f^*_{k-1} достигается в трех соседних точках

$$x_{k_1}, \ x_{k_2}, \ x_{k_3}, \ k_1 \in [1 : (k-1)], \ k_2 \in [1 : (k-1)], \ k_3 \in [1 : (k-1)],$$

таких, что $x_{k_1} < x_{k_2} < x_{k_3}$, то прекратить вычисления (в этом случае все точки отрезка $[x_{k_1}, x_{k_3}]$ являются решением задачи 1).

(Шаги VII—IX соответствуют той ситуации, когда внутри интер-

вала локализации $[x_-, x_+]$ точки минимума не проводилось вычисление функции f_0).

VII. Если $k = N + 1$, то прекратить вычисления (в этом случае точка минимума функции f_0 принадлежит интервалу $[x_-, x_+]$, длина которого не превышает гарантированного значения $(b_0 - a_0)/F_N$); иначе перейти к шагу VIII.

VIII. Вычислить точку x_k следующего вычисления функции f_0

$$x_k = x_- + (x_+ - x_-) F_{N-k+1}/F_{N-k+2}.$$

IX. Вычислить $f_0(x_k)$, положить $k = k + 1$ и перейти к шагу IV.

(Шаги X—XX соответствуют той ситуации, когда внутри интервала локализации $[x_-, x_+]$ точки минимума существует точка $\overset{*}{x}_{k-1}$, в которой на предыдущих итерациях проводилось вычисление функции f_0).

X. Если справа от точки $\overset{*}{x}_{k-1}$ существуют по крайней мере две точки, в которых функция f_0 вычислялась на предыдущих итерациях, то вычислить точку

$$x_+ = x_{k_4} + \frac{\overset{*}{f}_{k-1} - f_0(x_{k_4})}{f_0(x_{k_5}) - f_0(x_{k_4})}(x_{k_5} - x_{k_4}),$$

где x_{k_4}, x_{k_5}, $k_4 \in [1 : (k-1)]$, $k_5 \in [1 : (k-1)]$, ближайшие справа к $\overset{*}{x}_{k-1}$ точки, такие, что $\overset{*}{x}_{k-1} < x_{k_4} < x_{k_5}$, и перейти к шагу XIII; иначе перейти к шагу XI.

XI. Если справа от точки $\overset{*}{x}_{k-1}$ существует только одна точка x_{k_4}, $k_4 \in [1 : (k-1)]$, $(\overset{*}{x}_{k-1} < x_{k_4})$, в которой функция f_0 вычислялась на предыдущих итерациях, то положить $x_+ = x_{k_4}$ и перейти к шагу XIII; иначе перейти к шагу XII.

XII. Если справа от точки $\overset{*}{x}_{k-1}$ не существует точек, в которых функция f_0 вычислялась (т. е., если $x_i \leqslant \overset{*}{x}_{k-1}$, $i = 1, \ldots, k-1$), то положить $x_+ = b_0$ и перейти к шагу XIII.

XIII. Если слева от точки $\overset{*}{x}_{k-1}$ существуют по крайней мере две точки, в которых функция f_0 вычислялась на предыдущих итерациях, то вычислить

$$x_- = x_{k_6} + \frac{\overset{*}{f}_{k-1} - f_0(x_{k_6})}{f_0(x_{k_7}) - f_0(x_{k_6})}(x_{k_7} - x_{k_6}),$$

где x_{k_6}, x_{k_7}, $k_6 \in [1 : (k-1)]$, $k_7 \in [1 : (k-1)]$, ближайшие слева к $\overset{*}{x}_{k-1}$ точки, такие, что $\overset{*}{x}_{k-1} > x_{k_6} > x_{k_7}$, и перейти к шагу XVI; иначе перейти к шагу XIV.

XIV. Если слева от точки $\overset{*}{x}_{k-1}$ существует только одна точка x_{k_6}, $k_6 \in [1 : (k-1)]$, $(\overset{*}{x}_{k-1} > x_{k_6})$, в которой функция f_0 вычислялась на предыдущих итерациях, то положить $x_- = x_{k_6}$ и перейти к шагу XVI; иначе перейти к шагу XV.

XV. Если слева от точки x^*_{k-1} не существует точек, в которых функция f_0 вычислялась (т. е., если $x_i \geqslant x^*_{k-1}$, $i = 1, \ldots, k-1$), то положить $x_- = a_0$ и перейти к шагу XVI.

XVI. Если $k = N + 1$, то прекратить вычисления (в этом случае точка минимума функции f_0 принадлежит интервалу $[x_-, x_+]$, длина которого не превышает гарантированного значения $(b_0 - a_0)/F_N$); иначе перейти к шагу XVII.

XVII. Вычислить

$$c_k = (x^*_{k-1} - x_-)/(x_+ - x_-).$$

XVIII. Вычислить

$$t_k = \begin{cases} 1 - (1 - c_k) F_{N-k+1}/F_{N-k+2}, & \text{если } 0 \leqslant c_k \leqslant 1/2; \\ c_k F_{N-k+1}/F_{N-k+2}, & \text{если } 1/2 \leqslant c_k \leqslant 1. \end{cases}$$

XIX. Вычислить точку x_k следующего вычисления функции f_0

$$x_k = x_- + (x_+ - x_-) t_k.$$

XX. Вычислить значение $f_0(x_k)$, положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Пусть выполняются предположения 1 и $[x_-, x_+]$ — интервал локализации точки минимума функции f_0, вычисленный в k-й итерации ($k = 1, \ldots, N$). Тогда за оставшиеся $N - k + 1$ итераций алгоритм 1 приводит либо к интервалу, каждая точка которого является решением задачи 1, либо к интервалу локализации точки минимума функции f_0, длина которого не превышает значения $(x_+ - x_-)s_{N-k+1}$, где*

$$s_{N-k+1} = \begin{cases} \max\{(1 - c_k)/F_{N-k+1}, c_k/F_{N-k}\}, & 0 \leqslant c_k \leqslant 1/2; \\ \max\{(1 - c_k)/F_{N-k}, c_k/F_{N-k+1}\}, & 1/2 \leqslant c_k \leqslant 1, \end{cases}$$

если внутри интервала локализации $[x_-, x_+]$ точки минимума вычисления функции f_0 проводились;

$$s_{N-k+1} = 1/F_{N-k+1},$$

если внутри интервала локализации $[x_-, x_+]$ точки минимума вычисления функции f_0 не проводились.

Библиографические указания. Параграф написан на основании работ [387, 389].

В работах [9, 10] предлагаются оптимальные алгоритмы поиска минимума функций с ограниченной второй и третьей производной. В работе [499] рассматривается алгоритм минимизации на заданном отрезке многоэкстремальных функций, являющихся суммой вогнутой и выпуклой функций (в этот класс входят, например, функции с ограниченной второй производной).

1.5. Методы типа Ньютона

Задача 0. Найти $\arg \min_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0: R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 0. Функция f_0 дважды непрерывно дифференцируемая на $[a_0, b_0]$ и $|f_0''(x) - f_0''(y)| \leqslant \alpha |x - y|$, $\alpha < \infty$.

1. Метод Ньютона

Идея метода Ньютона заключается в том, что функция $f_0'(x)$ линеаризуется в окрестности x^k и находят точку x^{k+1}, в которой линеаризованная функция превращается в ноль.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in [a_0, b_0]$; положить $k = 0$.

Основной цикл. II. Вычислить $f_0'(x^k)$ — первую производную функции f_0 в точке x^k.

III. Если $f_0'(x^k) = 0$, то прекратить вычисления (в этом случае точка x^k является стационарной точкой функции f_0), иначе перейти к шагу IV.

IV. Вычислить $f_0''(x^k)$ — вторую производную функции f_0 в точке x^k.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \frac{f_0'(x^k)}{f_0''(x^k)}.$$

VI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Если выполнено предположение 0 и, кроме того, (i) — $-f_0'(a_0) f_0'(b_0) < 0$; (ii) — для всех $x \in [a_0, b_0]$ — $f_0''(x) > 0$; (iii) — для всех $x \in [a_0, b_0]$ выполняется неравенство*

$$0 \leqslant 1 - \left[\frac{f_0'(x)}{f_0''(x)}\right]' \leqslant \gamma < 1,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к стационарной точке $\bar{x}$ функции f_0 с квадратичной скоростью, т. е.

$$|x^{k+1} - \bar{x}| \leqslant \beta |x^k - \bar{x}|^2.$$

Здесь

$$\beta = \sup_{k=0,1,\ldots} \frac{|f_0'''(\theta^k) \cdot f_0''(\eta^k)|}{[f_0''(\theta^k)]^2},$$

где $\theta^k \in [x^k, \bar{x}]$; $\eta^k \in [\theta^k, \bar{x}]$.

2. Метод секущей

Метод секущей является модификацией метода Ньютона (алгоритма 1), в котором вместо второй производной $f''(x^k)$ используется ее разностная аппроксимация.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in [a_0, b_0]$;

положить $k = 0$.

Основной цикл. II. Вычислить $f_0'(x^k)$ — первую производную функции f_0 в точке x^k.

III. Если $f_0'(x^k) = 0$, то прекратить вычисления (в этом случае точка x^k является стационарной точкой функции f_0), иначе перейти к шагу IV.

IV. Вычислить следующее приближение

$$x^{k+1} = x^k - \frac{f_0'(x^k)(x^k - x^{k-1})}{f_0'(x^k) - f_0'(x^{k-1})}.$$

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Если выполнены все условия теоремы 1, то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, сходится к стационарной точке $\bar{x}$ функции f_0 со сверхлинейной скоростью*

$$\lim_{k\to\infty} \frac{|x^{k+1} - \bar{x}|}{|x^k - \bar{x}|^{\tau}} = \left| \frac{2f_0''(\bar{x})}{f_0'''(\bar{x})} \right|^{1/\tau},$$

где τ — решение уравнения $t^2 = t + 1$ ($\tau = (1 + \sqrt{5})/2 \cong 1,618$).

Библиографические указания. При написании параграфа использовались работы [425, 49].

1.6 Методы касательных

Задача 0. Найти $\arg \min\limits_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0: R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

1. Случай дифференцируемой функции

Предположение 1. (*i*) — функция f_0 непрерывно дифференцируемая на отрезке $[a_0, b_0]$; (*ii*) — функция f_0 выпукла на отрезке $[a_0, b_0]$; (*iii*) — $f_0'(a_0) < 0$ и $f_0'(b_0) > 0$.

Алгоритм 1

Начало. I. Выбрать число $\varepsilon > 0$ — точность вычисления точки минимума; положить $k = 0$.

II. Вычислить производные $f_0'(a_0)$ и $f_0'(b_0)$ функции f_0 в точках a_0 и b_0, соответственно.

Основной цикл. III. Найти точку x^k — корень уравнения

$$f_0(a_k) + f_0'(a_k)(x - a_k) = f_0(b_k) + f_0'(b_k)(x - b_k).$$

IV. Вычислить $f_0'(x^k)$.

V. Если $f_0'(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VI.

VI. Если $f_0'(x^k) < 0$, то положить $a_{k+1} = x^k$, $b_{k+1} = b_k$ и перейти к шагу VIII; иначе перейти к шагу VII.

VII. Если $f_0'(x^k) > 0$, то положить $a_{k+1} = a_k$, $b_{k+1} = x^k$ и перейти к шагу VIII.

VIII. Если $b_{k+1} - a_{k+1} \leqslant \varepsilon$, то положить $x^* = (a_{k+1} + b_{k+1})/2$ и прекратить вычисления; иначе перейти к шагу IX.

IX. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Если выполняются предположения 1, то для последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, справедливо*

$$\lim_{\substack{k\to\infty \\ \varepsilon\to 0}} f_0(x^k) = \min_{x\in[a_0,b_0]} f_0(x).$$

Если, кроме того, точка минимума x^* единственная, то

$$\lim_{\substack{k\to\infty \\ \varepsilon\to 0}} x^k = x^*.$$

2. Случай недифференцируемой функции

Предположение 2. (*i*) — функция f_0 выпукла на отрезке $[a_0, b_0]$; (*ii*) — $f_0'(a_0 + 0) < 0$, $f_0'(b_0 - 0) < 0$.

Алгоритм 2

Начало. I. Задать число $\varepsilon > 0$ — точность вычисления точки минимума функции f_0; положить $k = 0$.

II. Вычислить правостороннюю $f_0'(a_0 + 0)$ и левостороннюю $f_0'(b_0 - 0)$ производные функции f_0 в точках a_0 и b_0 соответственно и положить $\gamma_0 = f_0'(a_0 + 0)$, $\beta_0 = f_0'(b_0 - 0)$.

Основной цикл. III. Найти точку x^k — корень уравнения

$$f_0(a_k) + \gamma_k(x - a_k) = f_0(b_k) + \beta_k(x - b_k).$$

IV. Вычислить правостороннюю $f_0'(x^k + 0)$ и левостороннюю $f_0'(x^k - 0)$ производные функции f_0 в точке x^k.

V. Если $f_0'(x^k + 0) f_0'(x^k - 0) \leqslant 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе выбрать любое число δ_k из отрезка $[f_0'(x^k - 0), f_0'(x^k + 0)]$ и перейти к шагу VI.

VI. Если $\delta_k < 0$, то положить $a_{k+1} = x^k$, $\gamma_{k+1} = \delta_k$, $b_{k+1} = b_k$, $\beta_{k+1} = \beta_k$ и перейти к шагу VII; иначе положить $a_{k+1} = a_k$, $\gamma_{k+1} = \gamma_k$, $b_{k+1} = x^k$, $\beta_{k+1} = \delta_k$ и перейти к шагу VII.

VII. Если $b_{k+1} - a_{k+1} \leqslant \varepsilon$, то положить $x^* = (a_{k+1} + b_{k+1})/2$ и прекратить вычисления; иначе перейти к шагу VIII.

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Если выполняются предположения 2, то для последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, справедливы утверждения теоремы 1.*

Библиографические указания. Параграф написан на основании работы [194].

1.7. Метод квадратичной аппроксимации

Задача 1. Найти $\arg\min_{x \in R^1} f_0(x)$ для заданной унимодальной функции $f_0: R^1 \to R^1$.

Сущность метода квадратичной аппроксимации заключается в следующем. По трем точкам функция f_0 аппроксимируется квадратной параболой, после чего находится точка минимума этой параболы. На следующем этапе аппроксимации используются три соседние точки, между которыми находится точка минимума функции f_0.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^1$, точность вычисления точки минимума $\varepsilon > 0$ и начальное смещение $\delta_0 > 0$.

II. Если $f_0(x^0 - \varepsilon) \geqslant f_0(x^0)$ и $f_0(x^0 + \varepsilon) \geqslant f_0(x^0)$, то положить $\bar{x}^* = x^0$ и прекратить вычисления; иначе перейти к шагу III.

III. Если $f_0(x^0) < f_0(x^0 - \varepsilon)$, то положить $\delta = \delta_0$ и перейти к шагу IV; если $f_0(x^0) < f_0(x^0 + \varepsilon)$, то положить $\delta = -\delta_0$ и перейти к шагу IV.

IV. Положить $k = 0$.

V. Положить $x^{k+1} = x^k + \delta$.

VI. Если $f_0(x^{k+1}) \leqslant f_0(x^k)$, то положить $\delta = 2\delta$, $k = k + 1$ и перейти к шагу V; иначе перейти к шагу VII.

VII. Если $f_0(x^{k+1}) > f_0(x^k)$ и $\delta > 0$, то положить $x_1 = x^{k-1}$, $x_2 = x^k$, $x_3 = x^{k+1}$ и перейти к шагу VIII.

Если $f_0(x^{k+1}) > f_0(x^k)$ и $\delta < 0$, то положить $x_1 = x^{k+1}$, $x_2 = x^k$, $x_3 = x^{k-1}$ и перейти к шагу VIII.

Примечание: интервал $[x_1, x_3]$ содержит точку минимума x^* функции f_0.

VIII. Если $x_3 - x_1 \leqslant \varepsilon$, то положить $\bar{x}^* = x_2$ и прекратить вычисления; иначе перейти к шагу IX.

IX. Найти точку x_4 по формуле

$$x_4 = x_2 + \frac{\delta(f_0(x_1) - f_0(x_3))}{4(f_0(x_1) - 2f_0(x_2) + f_0(x_3))}.$$

Основной цикл. X. Если $x_4 < x_2$, то:

1) при $f_0(x_4) = f_0(x_2)$ и $f_0(x_1) > f_0(x_3)$ положить $x_1 = x_4$, $x_2 = x_2$, $x_3 = x_3$ и перейти к шагу XI;

2) при $f_0(x_4) = f_0(x_2)$ и $f_0(x_1) < f_0(x_3)$ положить $x_1 = x_1$, $x_3 = x_2$, $x_2 = x_4$ и перейти к шагу XI;

3) при $f_0(x_4) < f_0(x_2)$ положить $x_1 = x_1$, $x_3 = x_2$, $x_2 = x_4$ и перейти к шагу XI;

4) при $f_0(x_4) > f_0(x_2)$ положить $x_1 = x_4$, $x_2 = x_2$, $x_3 = x_3$ и перейти к шагу XI.

Если $x_4 > x_2$, то:

1) при $f_0(x_4) = f_0(x_2)$ и $f_0(x_1) < f_0(x_3)$ положить $x_1 = x_1$, $x_2 = x_2$, $x_3 = x_4$ и перейти к шагу XI;

2) при $f_0(x_2) = f_0(x_4)$ и $f_0(x_1) > f_0(x_3)$ положить $x_1 = x_2$, $x_2 = x_4$, $x_3 = x_3$ и перейти к шагу XI;

3) при $f_0(x_4) < f_0(x_2)$ положить $x_1 = x_2$, $x_2 = x_4$, $x_3 = x_3$ и перейти к шагу XI;

4) при $f_0(x_4) > f_0(x_2)$ положить $x_1 = x_1$, $x_2 = x_2$, $x_3 = x_4$ и перейти к шагу XI.

XI. Если $x_3 - x_1 \leqslant \varepsilon$, то положить $\bar{x}^* = x_2$ и прекратить вычисления; иначе перейти к шагу XII.

XII. Вычислить точку x_4 по формуле

$$x_4 = -\frac{1}{2}\,\frac{(x_2^2 - x_3^2)\,f_0(x_1) + (x_3^2 - x_1^2)\,f_0(x_2) + (x_1^2 - x_2^2)\,f_0(x_3)}{(x_3 - x_2)\,f_0(x_1) + (x_1 - x_3)\,f_0(x_2) + (x_2 - x_1)\,f_0(x_3)}$$

и перейти к шагу X (точка x_4 является точкой минимума квадратичной функции, проведенной через три точки x_1, x_2, x_3).

Замечание 1. Алгоритм, определяемый шагами I—IX, называют алгоритмом Дэвиса, Свенна, Кэмпи (ДСК). Алгоритм, определяемый шагами X—XII, соответствует алгоритму Пауэлла. Поэтому алгоритм 1 в [378] называется комбинированным алгоритмом Дэвиса, Свенна, Кэмпи — Пауэлла.

Теорема 1. *Если функция f_0 унимодальна, то за конечное число шагов алгоритм 1 приводит в точку $\bar{x}^*$, лежащую в ε-окрестности точки x^* (здесь x^* решение задачи 1).*

Библиографические указания. При написании параграфа использовались работы [378, 425].

1.8. Метод отыскания абсолютного минимума функций, удовлетворяющих условию Липшица

Задача 1. Найти $\arg\min\limits_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0: R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 1. Функция f_0 непрерывна на $[a_0, b_0]$ и удовлетворяет условию Липшица с константой γ, т. е.

$$|f_0(x') - f_0(x'')| \leqslant \gamma\,|x' - x''|, \quad \forall\, x',\ x'' \in [a_0, b_0].$$

На k-й итерации приводимого здесь алгоритма строится ломаная $\varphi_k(x, x^1, \ldots, x^k)$, ограничивающая $f_0(x)$ снизу, и вычисляются два числа f_+^k и f_-^k, которые являются, соответственно, верхней и нижней границами абсолютного минимума функции f_0 на отрезке $[a_0, b_0]$, такие, что

$$f_-^{k-1} \leqslant f_-^k \leqslant \min_{x \in [a_0, b_0]} f_0(x) \leqslant f_+^k \leqslant f_+^{k-1}, \quad k = 2, 3, \ldots,$$

причем, для каждого $\varepsilon > 0$ существует номер k такой, что

$$f_+^k - f_-^k \leqslant \varepsilon.$$

Число f_+^k является ординатой наиболее низкой из «верхних» вершин ломаной φ_k, а f_-^k — ординатой наиболее низкой из «нижних»

вершин ломаной φ_k. Абсциссы точек ломаной $\varphi_k(x, x^1, \ldots, x^k)$, лежащие ниже прямой $y = f_+^k$, образуют множество, содержащее множество решений X_* задачи 1. Начальное приближение в алгоритме произвольно.

По существу вычисления на каждой итерации сводятся лишь к решению двух линейных уравнений и выбору минимальных величин из конечного набора.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^1 \in [a_0, b_0]$ (обычно выбирается $x^1 = (a_0 + b_0)/2$).

II. Выбрать величину $\varepsilon > 0$ (точность вычисления минимума функции f_0 по функционалу).

III. Положить $k = 1$.

Основной цикл. IV. Вычислить $f_0(x^k)$.

V. Определить функцию $g_k(x, x^k)$, $(x \in [a_0, b_0])$ по правилу

$$g_k(x, x^k) = f_0(x^k) - \gamma |x - x^k|. \tag{1.3}$$

VI. Определить функцию $\varphi_k(x, x^1, \ldots, x^k)$, $(x \in [a_0, b_0])$ по правилу

$$\varphi_k = (x, x^1, \ldots, x^k) = \max_{i=1,\ldots,k} g_i(x, x^i)$$

и найти множество Q_k «нижних» вершин ломаной $\varphi_k(x, x^1, \ldots, x^k)$ на отрезке $[a_0, b_0]$:

если $k = 1$, то

$$Q_k = \{(a_0, f_0(x^1) - \gamma(x^1 - a_0)), (b_0, f_0(x^1) - \gamma(b_0 - x^1))\};$$

если $k > 1$, то

$$Q_k = (Q_{k-1} \setminus \{(x^k, \varphi_{k-1}(x^k, x^1, \ldots, x^{k-1}))\}) \cup G_k,$$

где G_k определяется:

если $x^k = a_0$, то G_k состоит из одной точки — точки пересечения прямых

$$\begin{aligned} y &= f_0(x^{k_2}) + \gamma(x - x^{k_2}), \\ y &= f_0(x^k) - \gamma(x - x^k), \end{aligned} \tag{1.4}$$

(здесь x^{k_2} — абсцисса ближайшей справа к $(x^k, f_0(x^k))$ «верхней» вершины ломаной $\varphi_k(x, x^1, \ldots, x^k)$);

если $x^k = b_0$, то G_k состоит из одной точки — точки пересечения прямых

$$\begin{aligned} y &= f_0(x^{k_1}) - \gamma(x - x^{k_1}), \\ y &= f_0(x^k) + \gamma(x - x^k), \end{aligned} \tag{1.5}$$

(здесь x^{k_1} — абсцисса ближайшей слева к $(x^k, f_0(x^k))$ «верхней» вершины ломаной $\varphi_k(x, x^1, \ldots, x^k)$);

если $x^k \neq a_0$ и $x^k \neq b_0$, то G_k состоит из двух точек — точки пересечения прямых (1.4) и точки пересечения прямых (1.5).

VII. Вычислить

$$f_+^k = \min_{i=1,\ldots,k} f_0(x^i).$$

VIII. Найти точку x^{k+1}, являющуюся точкой абсолютного минимума функции $\varphi_k(x, x^1, \ldots, x^k)$ на отрезке $[a_0, b_0]$

$$\varphi_k(x^{k+1}, x^1, \ldots, x^k) = \min_{a_0 \leqslant x \leqslant b_0} \varphi_k(x, x^1, \ldots, x^k), \qquad (1.6)$$

и положить $f_-^k = \varphi_k(x^{k+1}, x^1, \ldots, x^k)$.

IX. Если выполняется условие

$$f_+^k - f_-^k \leqslant \varepsilon,$$

то прекратить вычисления (в этом случае абсциссы точек ломаной $\varphi_k(x, x^1, \ldots, x^k)$, лежащие ниже прямой $y = f_+^k$, образуют множество X_ε, заведомо содержащее решения x^* задачи 1, и, кроме того, выполняется

$$f_-^k \leqslant f_0(x^*) \leqslant f_+^k);$$

иначе положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Пусть выполняется предположение 1, тогда алгоритм 1 порождает последовательность чисел* f_+^k, $k = 1, 2, \ldots$ *и* f_-^k, $k = 1, 2, \ldots$ *такие, что*

$$f_-^k \leqslant \min_{x \in [a_0, b_0]} f_0(x) \leqslant f_+^k, \quad k = 1, 2, \ldots$$

и, кроме того, при $\varepsilon = 0$ *имеют место предельные соотношения*

$$\lim_{k \to \infty} f_-^k = \lim_{k \to \infty} f_+^k = \min_{x \in [a_0, b_0]} f_0(x) = f_0(x^*).$$

Библиографические указания. При написании параграфа использовались работы [281, 107].

В работе [202] для специального класса многоэкстремальных функций, не являющихся вогнутыми, строится оптимальный алгоритм по критерию, равному максимальной возможной величине ошибки в определении экстремума минимизируемой функции.

1.9. Метод кусочно-кубической аппроксимации

Задача 1. Найти $\arg \min_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0: R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 1. Функция f_0 удовлетворяет условию Липшица на отрезке $[a_0, b_0]$.

В методе кусочно-кубической аппроксимации в качестве модели функции $f_0(x)$ используется кусочно-кубическая кривая $\varphi_k(x)$, проходящая через точки $(x^j, f_0(x^j))$, $j \in [0 : N_k]$, и моделирующая поведение функции $f_0(x)$ полиномами третьей степени на отдельных участках отрезка $[a_0, b_0]$. Отыскание приближенного значения глобального минимума функции $f_0(x)$ сводится к вычислению минимумов кубических парабол на отдельных интервалах отрезка $[a_0, b_0]$ (т. е. к решению соответствующих квадратных уравнений).

Алгоритм 1

Начало. I. Выбрать константу $\varepsilon > 0$, характеризующую относительную ошибку аппроксимации функции $f_0(x)$ кусочно-кубической кривой.

II. Положить $N_0 = 5$.

III. Положить $k = 0$.

IV'. Вычислить точки x^i, $i \in [0 : N_0 - 1]$ по формулам

$$x^i = a_0 + \frac{b_0 - a_0}{N_0 - 1} i, \quad i \in [0 : N_0 - 1]$$

и значение функции f_0 в этих точках.

Основной цикл. IV. Положить $j = 0$.

V. Провести кубическую параболу $\psi_{j+1}(x)$ через четыре точки

$(x^j, f_0(x^j))$, $(x^{j+1}, f_0(x^{j+1}))$, $(x^{j+2}, f_0(x^{j+2}))$, $(x^{j+3}, f_0(x^{j+3}))$.

VI. Если $j = 0$, то на отрезке $[x^0, x^2]$ аппроксимирующую кусочно-кубическую кривую $\varphi_k(x)$ определить по правилу

$$\varphi_k(x) = \psi_1(x), \quad x \in [x^0, x^2]$$

и перейти к шагу IX; иначе перейти к шагу VII.

VII. Если $j + 3 < N_k - 1$, то на отрезке $[x^{j+1}, x^{j+2}]$ аппроксимирующую кусочно-кубическую кривую $\varphi_k(x)$ определить по правилу

$$\varphi_k(x) = \psi_{j+1}(x), \quad x \in [x^{j+1}, x^{j+2}]$$

и перейти к шагу IX; иначе перейти к шагу VIII.

VIII. Если $j + 3 = N_k - 1$, то на отрезке $[x^{N_k-3}, x^{N_k-1}]$ аппроксимирующую кусочно-кубическую кривую $\varphi_k(x)$ определить по правилу

$$\varphi_k(x) = \psi_{j+1}(x), \quad x \in [x^{N_k-3}, x^{N_k-1}]$$

и перейти к шагу X; иначе перейти к шагу X;

IX. Положить $j = j + 1$ и перейти к шагу V.

X. Положить $N_{k+1} = 2N_k - 1$.

XI. Вычислить точки

$$x^i = a_0 + \frac{b_0 - a_0}{N_{k+1} - 1} i, \quad i \in [0 : N_{k+1} - 1].$$

XII. Если выполняется неравенство

$$\max_{0 \leqslant i \leqslant N_{k+1}-1} |(f_0(x^i) - \varphi_k(x^i))/f_0(x^i)| \leqslant \varepsilon,$$

то перейти к шагу XIII; иначе положить $k = k + 1$ и перейти к шагу IV.

XIII. Положить $j = 0$.

XIV. Если $j = 0$, то вычислить точку минимума $\bar{x}^1$ кубической параболы $\psi_1(x)$ на отрезке $[x^0, x^2]$ и перейти к шагу XVII; иначе перейти к шагу XV.

XV. Если $j+3 < N_k - 1$, то вычислить точку минимума $\bar{x}^{j+1}$ кубической параболы $\psi_{j+1}(x)$ на отрезке $[x^{j+1}, x^{j+2}]$ и перейти к шагу XVII; иначе перейти к шагу XVI.

XVI. Вычислить точку минимума $\bar{x}^{N_k-3}$ кубической параболы $\psi_{N_k-3}(x)$ на отрезке $[x^{N_k-3}, x^{N_k-1}]$ и перейти к шагу XVIII.

XVII. Положить $j = j + 1$ и перейти к шагу XV.

XVIII. Найти точку x^ε (приближенное решение), принадлежащую множеству точек

$$\{x^j,\ j\in[0:N_{k+1}-1];\ \bar{x}^j,\ j\in[1:N_k-3]\},$$

такую, что

$$f_0(x^\varepsilon) = \min\{f_0(x^j),\ j\in[0:N_{k+1}-1];\ f_0(\bar{x}^j),\ j\in[1:N_k-3]\}.$$

Библиографические указания. Параграф написан на основании работ [378, 17].

В работе [472] предложен метод, использующий квадратичную и кубическую интерполяцию и не требующий вычисления производных минимизируемой функции.

1.10. Методы глобального поиска

Задача 0. Найти $\arg\min\limits_{x\in[a_0,b_0]} f_0(x)$ для заданной функции $f_0 : R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$ вещественной оси R^1.

Приводимые ниже алгоритмы применяют для вычисления абсолютного минимума многоэкстремальной функции f_0 на отрезке $[a_0, b_0]$.

1. Алгоритм глобального поиска

Алгоритм 1

Начало. I. Выбрать произвольную константу $\alpha > 1$

II. Положить $x^0 = a_0$, $x^1 = b_0$.

III. Положить $k = 1$.

Основной цикл. IV. Разместить точки последовательности $\{x^i\}_{i=0}^k$ в порядке возрастания их значений и обозначить новую последовательность через $\{\bar{x}^i\}_{i=0}^k$, т. е.

$$a_0 = \bar{x}^0 < \bar{x}^1 < \cdots < \bar{x}^k = b_0.$$

V. Найти максимальное абсолютное значение относительной первой разности

$$\delta_k = \max_{1\le i\le k} |(f_0(\bar{x}^i) - f_0(\bar{x}^{i-1}))/(\bar{x}^i - \bar{x}^{i-1})|.$$

VI. Если $\delta_k = 0$, то положить $\beta_k = 1$ и перейти к шагу VII; иначе положить $\beta_k = \alpha\delta_k$ и перейти к шагу VII.

VII. Положить $i = 1$.

VIII. Вычислить $\gamma(i)$ — характеристику интервала $(\bar{x}^{i-1}, \bar{x}^i)$

$$\gamma(i) = \beta_k(\bar{x}^i - \bar{x}^{i-1}) + \frac{(f_0(\bar{x}^i) - f_0(\bar{x}^{i-1}))^2}{\beta_k(\bar{x}^i - \bar{x}^{i-1})} - 2(f_0(\bar{x}^i) + f_0(\bar{x}^{i-1})).$$

IX. Если $i < k$, то положить $i = i + 1$ и перейти к шагу VIII; иначе перейти к шагу X.

X. Найти наименьшее значение $i_k \in [1 : k]$, при котором выполняется равенство

$$\gamma(i_k) = \max_{1 \leqslant i \leqslant k} \gamma(i).$$

XI. Вычислить следующее приближение

$$x^{k+1} = (\bar{x}^{i_k} + \bar{x}^{i_k-1})/2 - (f_0(\bar{x}^{i_k}) - f_0(\bar{x}^{i_k-1}))/2\beta_k.$$

XII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Пусть функция $f_0(x)$ удовлетворяет на $[a_0, b_0]$ условию Липшица с константой $\gamma < \infty$*

$$|f_0(x') - f_0(x'')| \leqslant \gamma |x' - x''|, \quad \forall x', x'' \in [a_0, b_0].$$

*Тогда: (**i**) — если функция $f_0(x)$ имеет на отрезке $[a_0, b_0]$ конечное число локальных экстремумов, то любая предельная точка $\bar{x}$ последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, локально-оптимальна; (**ii**) — если наряду с предельной точкой $\bar{x}$ существует другая предельная точка $\hat{x}$ последовательности $\{x^k\}_{k=0}^{\infty}$, то $f_0(\bar{x}) = f_0(\hat{x})$; (**iii**) — если $\bar{x}$ — предельная точка последовательности $\{x^k\}_{k=0}^{\infty}$, то $f_0(x^k) \geqslant f_0(\bar{x})$, $k = 0, 1, \ldots$; (**iv**) — если на некоторой итерации алгоритма 1 справедливо неравенство $\beta_k > 2\gamma$, то множество предельных точек последовательности $\{x^k\}_{k=0}^{\infty}$ совпадает с множеством точек абсолютного минимума функции f_0 на отрезке $[a_0, b_0]$.*

Теорема 1'. *Пусть для наперед выбранной константы $\varepsilon > 0$ (ε — точность вычисления абсолютного минимума) при помощи алгоритма 1 построена последовательность точек $x^0, x^1, \ldots, x^{k(\varepsilon)}$, где $k(\varepsilon)$ — наименьший индекс k, при котором выполняется неравенство*

$$\bar{x}^{i_k}(\varepsilon) - \bar{x}^{i_{k(\varepsilon)}-1} \leqslant \varepsilon.$$

*Тогда: (**i**) — если в $(k(\varepsilon) + 1)$-й итерации выполняется неравенство*

$$\beta_{k(\varepsilon)} \geqslant \gamma(2\mu + 1)/(\mu - 1),$$

где $\mu = \min_{i \in \mathcal{I}} \frac{\bar{x}^i - \bar{x}^{i-1}}{\varepsilon}$; $\mathcal{I} = \{i \,|\, \bar{x}^i - \bar{x}^{i-1} > \varepsilon, \ 1 \leqslant i \leqslant k(\varepsilon)\}$, *то*

$$f_0(x) \geqslant \min_{0 \leqslant i \leqslant k(\varepsilon)} f_0(x^i), \quad x \in [\bar{x}^{i-1}, \bar{x}^i], \ i \in \mathcal{I},$$

т. е. точка x^* абсолютного минимума не может принадлежать интервалу, длина которого превышает заданную точность ε; (ii) — если $\alpha > \sqrt{\mu}/(\sqrt{\mu} - 1)$, то

$$\min\{f_0(\bar{x}^i),\ f_0(\bar{x}^{i-1})\} > \min_{0 \le j \le k(\varepsilon)} f_0(x^j), \quad i \in \mathcal{I},$$

т. е. оценка $\min\limits_{0 \le j \le k(\varepsilon)} f_0(x^j)$ минимального значения функции f_0 достигается на одном из концов интервала, длина которого не превышает точности ε; (iii) — для любого положительного δ существует столь большое значение коэффициента α, что точки $x^0, x^1, \ldots, x^{k(\varepsilon)}$ образуют $(\varepsilon + \delta)$-сеть в интервале $[a_0, b_0]$.

Замечание 1. Значение константы α должно быть достаточно большим, чтобы удовлетворялось неравенство $\beta_k > 2\gamma$, которое является достаточным условием сходимости алгоритма 1. Но, с другой стороны, с ростом α возрастает (приближаясь при $\alpha \to \infty$ к количеству узлов сетки метода перебора) число вычислений функции f_0. В случае, когда известна константа Липшица γ, выбор значения α не вызывает затруднений. В случае, если известны лишь грубые априорные верхние оценки константы Липшица, можно воспользоваться описанным ниже алгоритмом.

2. Рандомизированный алгоритм глобального поиска

Алгоритм 2

Начало. I. Выбрать нижнюю α_1 и верхнюю α_2 оценки параметра α.

II—X. Шаги II—X такие, как в алгоритме 1, с тем лишь отличием, что при вычислении β_k на шаге VI вместо константы α следует взять константу α_1.

XI. Вычислить

$$\nu_1 = \beta_k(\bar{x}^{i_k} - \bar{x}^{i_k - 1}) - (f_0(\bar{x}^{i_k}) + f_0(\bar{x}^{i_k - 1})).$$

XII. Положить $j_1 = i_k$, $\gamma_1 = \gamma(i_k)$.

XIII. Если $\delta_k = 0$, то положить $\beta'_k = 1$ и перейти к шагу XIV; иначе положить $\beta'_k = \alpha_2\delta_k$ и перейти к шагу XIV.

XIV. Положить $i = 1$.

XV. Вычислить

$$\gamma(i) = \beta'_k(\bar{x}^i - \bar{x}^{i-1}) + \frac{(f_0(\bar{x}^i) - f_0(\bar{x}^{i-1}))^2}{\beta'_k(\bar{x}^i - \bar{x}^{i-1})} - 2(f_0(\bar{x}^i) - f_0(\bar{x}^{i-1})).$$

XVI. Если $i < k$, то положить $i = i + 1$ и перейти к шагу XV; иначе перейти к шагу XVII.

XVII. Найти наименьшее значение индекса $i'_k \in [1 : k]$, при котором выполняется равенство $\gamma(i'_k) = \max\limits_{1 \le i \le k} \gamma(i)$.

XVIII. Вычислить

$$\nu_2 = \beta'_k(\bar{x}^{i'_k} - \bar{x}^{i'_k - 1}) - (f_0(\bar{x}^{i'_k}) + f_0(\bar{x}^{i'_k - 1})).$$

XIX. Положить $j_2 = i'_k$, $\gamma_2 = \gamma(i'_k)$.

XX. Если $j_1 = j_2$, то вычислить $p = \alpha_1/(\alpha_1 + \alpha_2)$ и перейти к шагу XXI; иначе вычислить $p = (\nu_1 + \nu_2 - \gamma_2)/(2\nu_1 + 2\nu_2 - \gamma_1 - \gamma_2)$ и перейти к шагу XXI.

XXI. Однократно реализовать случайный механизм с двумя исходами 1 и 2, соответственно, имеющими вероятности $p_1 = p$ и $p_2 = 1 - p$.

XXII. Если случайный механизм дал исход 1, то положить $\beta = \beta_k$, $j = j_1$ и перейти к шагу XXIII; иначе положить $\beta = \beta'_k$, $j = j_2$ и перейти к шагу XXIII.

XXIII. Вычислить следующее приближение

$$x^{k+1} = (\bar{x}^j + \bar{x}^{j-1})/2 - (f_0(\bar{x}^j) - f_0(\bar{x}^{j-1}))/(2\beta).$$

XXIV. Положить $k = k + 1$ и перейти к шагу IV.

Библиографические указания. При написании параграфа использовались работы [343—346].

1.11 Методы поиска интервала наибольших значений многоэкстремальных функций

Задача 1. Найти $\arg \max\limits_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0 : R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Определение 1. Зафиксируем число $\delta > 0$ и определим класс K_δ функций, удовлетворяющих условию: для любой $\varphi(x)$ из K_δ существует отрезок Δ длиной δ из $[a_0, b_0]$ такой, что из $x \in \Delta$, $x' \notin \Delta$ следует $\varphi(x) > \varphi(x')$. Определение класса функций L_δ вытекает из определения класса K_δ, если положить $\delta = (b_0 - a_0)/2$.

Предположение 1. Функция f_0 непрерывна на $[a_0, b_0]$ и принадлежит классу K_δ.

Ниже приводится метод для отыскания отрезка Δ для функции $f_0(x)$, принадлежащей к классу K_δ. Если вычислить значения функции $f_0(x)$ из класса K_δ в точках $a_0 + \delta$, $a_0 + 2\delta$, ..., $a_0 + m\delta$, где

$$m = \begin{cases} (b_0 - a_0)/\delta - 1, & \text{если } (b_0 - a_0)/\delta \text{ — целое число;} \\ \mathrm{Ent}\,((b_0 - a_0)/\delta), & \text{если } (b_0 - a_0)/\delta \text{ — нецелое число,} \end{cases}$$

и среди этих точек выбрать точку d с наибольшим значением функции $f_0(d)$, то точка d принадлежит Δ, а область Δ следует искать на отрезке $[d - \delta, d + \delta]$. Таким образом, задача отыскания области Δ наибольших значений функции $f_0(x)$ из класса K_δ сводится к задаче отыскания области Δ для функции из класса L_δ. Если наибольшее значение достигается в двух соседних точках $a_0 + i\delta$, $a_0 + (i + 1)\delta$, то искомая область Δ совпадает с интервалом $[a_0 + i\delta, a_0 + (i + 1)\delta]$.

Приводимый ниже алгоритм для функции $f_0(x)$, принадлежащей классу L_δ на отрезке $[d - \delta, \delta + d]$, за $(n - 1)$ итераций находит Δ отрезок с заданной погрешностью $\varepsilon > 0$ за наименьшее воз-

можное число итераций. Для начала работы алгоритма необходимо знать некоторый интервал $[\alpha_0^+, \beta_0^+]$, принадлежащий отрезку Δ и содержащий точку d.

Алгоритм 1

Начало. I. Выбрать число $\varepsilon > 0$ — погрешность вычисления области Δ.

II. Задать отрезок $\Delta_0 = [\alpha_0^+, \beta_0^+]$, который содержит точку d и принадлежащий области Δ.

III. Вычислить натуральное число n, удовлетворяющее условию

$$F_{n-1} < (\delta - \delta_0)\frac{1}{\varepsilon} \leqslant F_n,$$

где F_{n-1}, F_n — числа Фибоначчи (т. е. такие, что $F_0 = F_1 = 1$, $F_i = F_{i-1} + F_{i-2}$, $i = 2, 3, \ldots$); $\delta_0 = \beta_0^+ - \alpha_0^+$.

IV. Положить $k = 0$, $G_0 = \varnothing$ (т. е. G_0 — пустое множество).

Основной цикл. V. Вычислить δ_k — длину интервала $[\alpha_k^+, \beta_k^+]$, принадлежащего отрезку Δ, и величину $\nu_k = \delta - \delta_k$.

VI. Найти точку α_k, лежащую слева от точки β_k^+ на расстоянии δ, и найти точку β_k, лежащую справа от точки α_k^+ на расстоянии δ.

VII. Если оба полуинтервалы $[\alpha_k, \alpha_k^+)$ и $(\beta_k^+, \beta_k]$ не содержат точек, в которых функция f_0 вычислялась на предыдущих итерациях, то вычислить значения функции $f_0(x)$ в точках

$$x'_k = \alpha_k^+ - \nu_k F_{n-k-2}/F_{n-k}; \quad x''_k = \beta_k^+ + \nu_k F_{n-k-2}/F_{n-k},$$

положить $G_{k+1} = G_k \cup \{x'_k\} \cup \{x''_k\}$ и перейти к шагу VIII; если полуинтервал $[\alpha_k, \alpha_k^+)$ содержит точку y'_k, в которой функция f_0 вычислялась на предыдущих итерациях, и полуинтервал $(\beta_k^+, \beta_k]$ не содержит точек, в которых функция f_0 вычислялась на предыдущих итерациях, то вычислить функцию f_0 в точке $x''_k = \beta_k - (y'_k - \alpha_k)$, положить $G_{k+1} = G_k \cup \{x''_k\}$ и перейти к шагу VIII; если полуинтервал $(\beta_k^+, \beta_k]$ содержит точку y''_k, в которой функция f_0 вычислялась на предыдущих итерациях, а полуинтервал $[\alpha_k, \alpha_k^+)$ не содержит таких точек, то вычислить функцию f_0 в точке $x'_k = \alpha_k + (\beta_k - y''_k)$, положить $G_{k+1} = G_k \cup \{x'_k\}$ и перейти к шагу VIII.

VIII. Используя имеющееся на k-й итерации множество точек G_{k+1}, значения функции f_0 в точках множества G_{k+1} и интервал $[\alpha_k^+, \beta_k^+]$, принадлежащий отрезку Δ, вычислить наибольший интервал $[\alpha_{k+1}^+, \beta_{k+1}^+]$, принадлежащий отрезку Δ, с помощью следующих пяти правил:

1) если x, $x' \in \Delta$ и $x \leqslant z \leqslant x'$, то $z \in \Delta$;
2) если $x \in \Delta$ и $f_0(x') \geqslant f_0(x)$, то $x' \in \Delta$;
3) если $x \notin \Delta$ и $f_0(x') \leqslant f_0(x)$, то $x' \notin \Delta$;
4) если $x \in \Delta$ и $|x - x'| > \delta$, то $x' \notin \Delta$;

5) если $x \in [d-\delta, d]$ и $x' \in [d, d+\delta]$, то из $f_0(x) \geqslant f_0(x')$ и $|x - x'| \leqslant \delta$ следует $x \in \Delta$, а из $f_0(x) \geqslant f_0(x')$ и $|x - x'| > \delta$ следует $x' \notin \Delta$.

IX. Если $k < n-2$, то положить $k = k+1$ и перейти к шагу V; иначе прекратить вычисления.

Теорема 1. *Если функция $f_0(x)$ принадлежит классу L_δ на интервале $[d-\delta, d+\delta]$, то алгоритм 1 за $(n-1)$ итераций находит интервал $[\alpha_{n-1}^+, \beta_{n-1}^+]$, принадлежащий отрезку Δ, такой, что $\beta_{n-1}^+ - \alpha_{n-1}^+ \geqslant \delta - \varepsilon$, причем алгоритм 1 является оптимальным алгоритмом поиска области Δ с погрешностью ε, т. е. находит область Δ с погрешностью ε за наименьшее возможное число итераций.*

Замечание 1. Если в алгоритме 1 на шаге VII точки x_k' и x_k'' вычислять по формулам

$$x_k' = \alpha_k + \nu_k F_{n-k-2}/F_{n-k}; \quad x_k'' = \beta_k - \nu_k F_{n-k-2}/F_{n-k},$$

то для этого алгоритма теорема 1 остается в силе.

Библиографические указания. Параграф написан на основании работ [350, 351].

1.12. Методы поиска глобального минимума, использующие стохастические автоматы

Задача 0. Найти $\arg\min_{x \in [a_0, b_0]} f_0(x)$ для заданной функции $f_0: R^1 \to R^1$ и заданного интервала $[a_0, b_0]$.

Предположение 0. Функция f_0 непрерывна на $[a_0, b_0]$ и имеет единственный глобальный минимум на $[a_0, b_0]$.

В начальной стадии метода интервал $[a_0, b_0]$ разбивается на m (достаточно большое число) подынтервалов l_j, $j = 1, \ldots, m$, равной длины $w = (b_0 - a_0)/m$ и каждому подынтервалу l_j, $j = 1, \ldots, m$, ставится в соответствие состояние S_j, $j = 1, \ldots, m$, стохастического автомата. Стохастический автомат задается вектором вероятностей $p(k) = (p_1(k), \ldots, p_m(k))$, каждая j-я компонента которого характеризует вероятность $p_j(k)$ перехода автомата в состояние S_j. На k-й итерации по заданному вектору $p(k)$ генерируется состояние автомата S_{i_k}, $i_k \in [1:m]$. Выход стохастического автомата принадлежит интервалу $[w(i_k - 1), wi_k]$ с равномерным распределением вероятностей. Вектор вероятностей $p(k)$ при переходе от одной итерации к другой преобразуется таким образом, что вероятность $p_i(k)$, соответствующая интервалу l_i, который содержит точку глобального минимума, возрастает, а все вероятности $p_j(k)$, $j = 1, \ldots, m$, $j \neq i$, уменьшаются. Приводимые ниже алгоритмы находят интервал l_{i_k}, $i_k \in [1:m]$, в котором с вероятностью, достаточно близкой к единице, находится точка глобального минимума функции f_0 на $[a_0, b_0]$.

Методы поиска глобального минимума, использующие стохастические автоматы, целесообразно применять тогда, когда вычисление функции $f_0(x)$ сопровождается сильными помехами.

1. Алгоритм, использующий модель Буша—Мостеллера

В алгоритме 1 вектор вероятностей $p(k)$ на каждой итерации преобразуется по формулам Буша — Мостеллера.

Алгоритм 1

Начало. I. Разбить интервал $[a_0, b_0]$ на m подынтервалов l_i равной длины $w = (b_0 - a_0)/m$ и поставить в соответствие каждому l_i, $i = 1, \ldots, m$, единственное состояние автомата S_i, $i = 1, \ldots, m$.

II. Положить $p_i(0) = 1/m$, $i = 1, \ldots, m$.

III. Положить $Q_i^* = \alpha_\infty$, $i = 1, \ldots, m$, где α_∞ — достаточно большое положительное число.

IV. Выбрать константу $\varepsilon > 0$ (ε — заданная точность выполнения условия оптимальности автомата в случайной среде).

V. Положить $k = 0$.

Основной цикл. VI. С помощью вектора вероятностей $p(k) = (p_1(k), p_2(k), \ldots, p_m(k))$ сгенерировать случайное состояние автомата $S(k) = S_{i_k}$, $i_k \in [1:m]$. Выбрать выход автомата $x_{i_k}(k)$, принадлежащий интервалу $[w(i_k - 1), wi_k]$ (считается, что случайная величина $x_{i_k}(k)$ равномерно распределена на этом интервале).

VII. Вычислить значение $f_0(x_{i_k})$.

VIII. Вычислить

$$Q_{\min}^* = \min_{1 \leqslant i \leqslant m} Q_i^*.$$

IX. Определить вход автомата $y(k+1)$ по правилам

$$y(k+1) = \begin{cases} 0, & \text{если} \quad Q_{\min}^* < f_0(x_{i_k}); \\ 1, & \text{если} \quad Q_{\min}^* \geqslant f_0(x_{i_k}). \end{cases}$$

X. Положить $Q_{i_k}^* = f_0(x_{i_k})$ (остальные значения Q_i^*, $i = 1, \ldots, m$, $i \neq i_k$, не меняются).

XI. Вычислить вектор $a(k) = (a_1(k), \ldots, a_m(k))$, определяющий структуру автомата на k-й итерации, по правилам:

если $y(k+1) = 1$, то $a_{i_k}(k) = 1$ и $a_j(k) = 0$ для $j = 1, 2, \ldots, m$, $j \neq i_k$;

если $y(k+1) = 0$, то $a_j(k) = p_j(k)$, $j = 1, \ldots, m$.

XII. Преобразовать вектор вероятностей $p(k)$ по формуле Буша — Мостеллера

$$p(k+1) = (\lambda I + (1-\lambda) A(k))\, p(k),$$

где I — единичная матрица размера $m \times m$; λ — постоянная величина, принадлежащая интервалу $[0, 1]$; $A(k)$ — $m \times m$-матрица, состоящая из m одинаковых столбцов $a(k)$.

XIII. Если с заданной точностью ε выполняются следующие условия оптимальности поведения автомата в случайной стационарной среде:

$$|p_{i_{k+1}}(k+1)-1|\leqslant\varepsilon, \text{ если } Q^*_{i_{k+1}}<Q^*_j \text{ для всех } j\neq i_{k+1}; \quad (1.7)$$

$$p_j(k+1)\leqslant\varepsilon \text{ для всех } j=1, \ldots, m, \quad j\neq i_{k+1}, \quad (1.8)$$

то прекратить вычисления (в этом случае точка глобального минимума с вероятностью, близкой к единице, принадлежит интервалу $l_{i_{k+1}}$); иначе положить $k=k+1$ и перейти к шагу VI.

При удачном разбиении интервала $[a_0, b_0]$ на подынтервалы l_j, $j=1, \ldots, m$, алгоритм 1 приводит к интервалу $[a_0+(i_{k+1}-1)w, a_0+i_{k+1}w)$, в котором с вероятностью, достаточно близкой к единице (за счет выбора ε), находится точка глобального минимума функции $f_0(x)$ на интервале $[a_0, b_0]$.

2. Алгоритм, использующий усредненные значения функции

В этом пункте приводится алгоритм, в котором вектор вероятностей $p(k)$ преобразуется по информации о средних значениях функции f_0 на соответствующих подынтервалах.

Алгоритм 2

Начало. I. Разбить интервал $[a_0, b_0]$ на m подынтервалов l_i равной длины $w=(b_0-a_0)/m$, и поставить в соответствие каждому l_i, $i=1, \ldots, m$, единственное состояние стохастического автомата S_i, $i=1, \ldots, m$.

II. Положить $p_i(0)=1/m$, $i=1, \ldots, m$.

III. Выбрать константу $\varepsilon>0$ (ε — заданная точность выполнения условия оптимальности стохастического автомата в случайной стационарной среде).

IV. Выбрать параметр алгоритма $\gamma>0$.

V. Вычислить массив средних значений

$$\bar z_i(0)=(f_0(w(i-{}^1/_2)))^{-\gamma}, \quad i=1, \ldots, m.$$

VI. Положить $k=0$.

Основной цикл. VII. С помощью вектора вероятностей $p(k)=(p_1(k), \ldots, p_m(k))$ сгенерировать случайное состояние автомата $S(k)=S_{i_k}$, $i_k\in[1:m]$. Выбрать выход автомата $x_{i_k}(k)$, принадлежащий интервалу $[w(i_k-1), wi_k]$ (считается, что случайная величина $x_{i_k}(k)$ равномерно распределена на интервале $[w(i_k-1), wi_k]$).

VIII. Вычислить значение $f_0(x_{i_k})$.

IX. Вычислить значение

$$z_{i_k}(k+1)=(f_0(x_{i_k}))^{-\gamma}.$$

X. Вычислить массив средних значений $\bar z_i(k+1)$, $i=1, \ldots, m$

по правилам:

$$\bar{z}_{i_k}(k+1) = \lambda \bar{z}_{i_k}(k) + (1-\lambda) z_{i_k}(k+1), \quad 0 < \lambda < 1;$$

$$\bar{z}_j(k+1) = \lambda \bar{z}_j(k), \quad j = 1, \ldots, m; \quad j \neq i_k.$$

XI. Вычислить вектор вероятностей $p(k+1) = (p_1(k+1), \ldots$ $\ldots, p_m(k+1))$ по формулам

$$p_j(k+1) = \bar{z}_j(k+1) / \sum_{i=1}^{m} \bar{z}_i(k+1), \quad j = 1, \ldots, m.$$

XII. Если с заданной точностью ε выполняются условия оптимальности (1.7), (1.8) поведения автомата в случайной стационарной среде, то прекратить вычисления; иначе положить $k = k + 1$ и перейти к шагу VII.

Алгоритм 2 обеспечивает выполнение условий (1.7), (1.8), если произведено удачное разбиение интервала $[a_0, b_0]$ на подынтервалы I_j, $j = 1, \ldots, m$, и выбран подходящий параметр алгоритма γ.

Библиографические указания. При написании параграфа использовались работы [17, 18].

1.13. Адаптивные методы

1. Алгоритмы Кифера—Вольфовица

З а д а ч а 1. Найти $\arg \max_{x \in R^1} f_0(x)$ для заданной функции $f_0 : R^1 \to R^1$.

Предположение 1. Функция f_0 унимодальна.

Метод Кифера — Вольфовица применяется для минимизации унимодальных функций, вычисление которых проводится со случайными помехами.

Алгоритм 1

Н а ч а л о. I. Выбрать произвольное начальное приближение $x^0 \in R^1$.

II. Положить $k = 0$.

О с н о в н о й ц и к л. III. Вычислить значения шагового множителя ρ_k и смещения δ_k, удовлетворяющие условиям теоремы 1.

IV. Найти величину $z(x^k + \delta_k)$ — результат вычисления со случайными помехами значения функции f_0 в точке $x^k + \delta_k$.

V. Найти величину $z(x^k - \delta_k)$ — результат вычисления со случайными помехами значения функции f_0 в точке $x^k - \delta_k$.

VI. Вычислить следующее приближение

$$x^{k+1} = x^k + (\rho_k / \delta_k)(z(x^k + \delta_k) - z(x^k - \delta_k)).$$

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть функция f_0 является унимодальной и удовлетворяет условию*

$$|f_0(x) - f_0(y)| < \alpha_1 |x - x^*| + \alpha_2 < \infty,$$

где x^* — решение задачи 1; α_1, α_2 — некоторые постоянные.

Тогда, если в алгоритме 1 ошибка вычислений функции f_0 равномерно ограничена и имеет нулевое математическое ожидание, т. е.

$$E(z(x^k \pm \delta_k) - f_0(x^k \pm \delta_k)) = 0;$$

$$\sum_{k=0}^{\infty} E(z(x^k \pm \delta_k) - f_0(x^k \pm \delta_k))^2 < \infty,$$

и если шаговые множители ρ_k и смещения δ_k такие, что

$$\rho_k > 0; \quad \lim_{k\to\infty} \rho_k = 0; \quad \lim_{k\to\infty} \delta_k = 0;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty; \quad \sum_{k=0}^{\infty} (\rho_k/\delta_k)^2 < \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к точке максимума x^* функции f_0 в среднеквадратическом и с вероятностью 1, т. е.

$$\lim_{k\to\infty} E(x^k - x^*)^2 = 0;$$

$$P\{\lim_{k\to\infty} x^k = x^*\} = 1.$$

Замечание 1. Если приближение x^{k+1} на шаге VI алгоритма 1 вычислять по формуле

$$x^{k+1} = x^k + \rho_k \operatorname{sign}\left\{\frac{z(x^k + \delta_k) - z(x^k - \delta_k)}{2\delta_k}\right\},$$

то получится нормализованный вариант метода Кифера — Вольфовица.

2. Простой перебор

З а д а ч а 2. Найти $\arg\min_{x\in[a_0,b_0]} f_0(x)$ для заданной функции $f_0 : R^1 \to R^1$ и заданного отрезка $[a_0, b_0]$.

Предположение 2. Функция f_0 такова, что на отрезке $[a_0, b_0]$ точка ее локального минимума x^* является точкой абсолютного минимума f_0 на отрезке $[a_0, b_0]$.

Алгоритм 2

Н а ч а л о. I. Задать: число $\varepsilon > 0$ — точность вычисления точки минимума функции f_0 на отрезке $[a_0, b_0]$; натуральное число N (рекомендуется выбирать число N из отрезка $[10^1, 10^3]$); положить $k = 0$.

О с н о в н о й ц и к л. II. Положить $j = 0$.

III. Положить $x_{k,0} = a_k$, вычислить $f_0(a_k)$ и положить $f_0(x_{k,0}) = f_0(a_k)$.

IV. Вычислить смещение

$$h_k = (b_k - a_k)/N.$$

V. Вычислить точку

$$x_{k,j+1} = x_{k,j} + h_k$$

и вычислить $f_0(x_{k,j+1})$.

VI. Если $f_0(x_{k,j+1}) > f_0(x_{k,j})$, то перейти к шагу VII; иначе перейти к шагу VIII.

VII. Если $j = 0$, то положить $a_{k+1} = a_k$, $b_{k+1} = x_{k,1}$ и перейти к шагу IX; иначе положить $a_{k+1} = x_{k,j-1}$, $b_{k+1} = x_{k,j+1}$ и перейти к шагу IX.

VIII. Если $j < N - 1$, то положить $j = j + 1$ и перейти к шагу V; иначе положить $a_{k+1} = x_{k,N-1}$, $b_{k+1} = b_k$ и перейти к шагу IX.

IX. Если $b_{k+1} - a_{k+1} > \varepsilon$, то перейти к шагу X; иначе положить $\bar{x}^* = (a_{k+1} + b_{k+1})/2$ и прекратить вычисления.

X. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Если выполнено предположение 2, то для произвольного $\varepsilon > 0$ алгоритм 2 за конечное число итераций приводит в точку $\bar{x}^*$ такую, что*

$$|\bar{x}^* - x^*| \leqslant \varepsilon.$$

Замечание 2. Недостатком метода простого перебора является то, что во многих «лишних» точках приходится вычислять значение функции f_0, что нежелательно в случае, когда f_0 определяется в результате эксперимента или когда для вычисления значения функции f_0 в точке требуется значительное машинное время.

Библиографические указания. При написании параграфа использовались работы [194, 358, 502].

Глава 2

МЕТОДЫ ОПТИМИЗАЦИИ ДИФФЕРЕНЦИРУЕМЫХ ФУНКЦИЙ

2.1. Градиентные методы

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположение 1. Функция f_0 дифференцируема в R^n.

В градиентных методах минимизации за направление движения в k-й итерации выбирается вектор, обратный градиенту функции f_0 в точке x^k. Различные варианты градиентного метода отличаются друг от друга способом выбора шагового множителя в k-й итерации, а также теми или иными способами (разностной) аппроксимации градиентов.

1. Метод наискорейшего спуска

В методе наискорейшего спуска шаговый множитель ρ_k ($k = 0, 1, \ldots$) вычисляется из условия минимума функции f_0 в направлении антиградиента $-\nabla f_0(x^k)$, т. е.

$$\rho_k = \arg\min_{\rho \geqslant 0} f_0(x^k - \rho \nabla f_0(x^k)).$$

Алгоритм 1

Начало. I. Выбрать произвольную начальную точку $x^0 \in R^n$ и положить $k = 0$.

Основной цикл. II. Вычислить $\nabla f_0(x^k)$.

III. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Вычислить шаговый множитель ρ_k из условия

$$f_0(x^k - \rho_k \nabla f_0(x^k)) = \min_{\rho \geq 0} f_0(x^k - \rho \nabla f_0(x^k)).$$

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \nabla f_0(x^k).$$

VI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Если выполнено предположение 1 и (i) — функция f_0 ограничена снизу в R^n; (ii) — градиент функции f_0 удовлетворяет условию Липшица*

$$\| \nabla f_0(x) - \nabla f_0(y) \| \leq \alpha \| x - y \|, \quad \forall x, y \in R^n, \quad \alpha < \infty,$$

то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, справедливо соотношение $\| \nabla f_0(x^k) \| \to 0$ при $k \to \infty$.

Теорема 1′. *Если выполнены условия: (i) — функция f_0 дважды непрерывно дифференцируема в R^n; (ii) — матрица вторых производных $\nabla^2_{xx} f_0(x)$ функции f_0 удовлетворяет условиям*

$$\beta \| y \|^2 \leq (\nabla^2_{xx} f_0(x) y, y) \leq \gamma \| y \|^2, \quad \gamma \geq \beta > 0,$$

при любых $x, y \in R^n$, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к точке минимума x^* со скоростью геометрической прогрессии со знаменателем $q = (\gamma - \beta)/(\gamma + \beta)$, т. е.

$$\| x^{k+1} - x^* \| \leq (\beta/\gamma)^{1/2} \| x^0 - x^* \| q^{k+1}.$$

Алгоритм 1 на практике используют редко. Это связано, в частности, с невозможностью реализовать шаг IV в большинстве практических случаев. Поэтому на практике чаще применяют модифицированные методы наискорейшего спуска.

2. Модифицированный метод наискорейшего спуска

Алгоритм 2

Шаги I—III такие, как в алгоритме 1.

IV. Определить число β_k, удовлетворяющее равенству

$$\beta_k = \min_{\beta \geq 0} f_0(x^k - \beta \nabla f_0(x^k)).$$

V. Вычислить параметр λ_k, удовлетворяющий условию (*iii*) теоремы 2.

VI. Вычислить шаговый множитель ρ_k из условия

$$f_0(x^k - \rho_k \nabla f_0(x^k)) \leq (1 - \lambda_k) f_0(x^k) + \lambda_k \beta_k. \qquad (2.1)$$

VII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \nabla f_0(x^k).$$

VIII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Если выполнено предположение 1 и (i) — градиент функции f_0 удовлетворяет условию Липшица*

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \alpha \|x - y\|, \quad \forall x, \ y \in R^n, \quad \alpha < \infty;$$

(ii) — начальное приближение x^0 таково, что

$$\sup_{x', x'' \in X_0} \|x' - x''\| = \operatorname{diam} X_0 = \eta < \infty,$$

где $X_0 = \{x \mid f_0(x) \leqslant f_0(x^0), \ x \in R^n\}$; (iii) — параметры λ_k, $k = 0, 1, \ldots$, удовлетворяют неравенствам $\lambda \leqslant \lambda_k \leqslant 1$, где λ — произвольная константа из полуинтервала (0, 1];

(iv) — функция f_0 выпукла в R^n, то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, справедлива оценка скорости сходимости по функционалу

$$f_0(x^k) - f_0^* \leqslant 2\alpha\eta^2 \left(\sum_{i=0}^{k-1} \lambda_i \right)^{-1} \leqslant 2\alpha\eta^2/\lambda k, \quad k = 1, 2, \ldots;$$

$$f_0^* \triangleq \min_{x \in R^n} f_0(x).$$

Теорема 2′. *Если выполнены все условия теоремы 2 и функция f_0 является сильновыпуклой с параметром сильной выпуклости $\nu > 0$, т. е.*

$$f_0(\theta x + (1 - \theta) y) \leqslant \theta f_0(x) + (1 - \theta) f_0(y) - \theta(1 - \theta) \nu \|x - y\|^2,$$

$$\forall x, \ y \in R^n, \quad 0 \leqslant \theta \leqslant 1,$$

то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, справедливы оценки скорости сходимости

$$f_0(x^k) - f_0^* \leqslant (f_0(x^0) - f_0^*) \exp(-\nu\lambda k/2\alpha), \quad k = 1, 2, \ldots;$$

$$\|x^k - x^*\|^2 \leqslant (2/\nu)(f_0(x^0) - f_0^*) \exp(-\nu\lambda k/2\alpha), \quad k = 1, 2, \ldots.$$

Замечание 2. Реализация алгоритма 2 на ЭВМ не приводит к искомому решению из-за наличия ошибок вычислений. Поэтому алгоритм 2 используют до тех пор, пока выполняется неравенство

$$(f_0(x^k) - f_0(x^{k+1}))/\|\nabla f_0(x^k)\|^2 \geqslant \varepsilon, \tag{2.2}$$

где $\varepsilon > 0$ выбранная на данном этапе точность вычислений.

Если же неравенство (2.2) нарушается, то следует либо повысить точность вычислений, либо перейти к другому, более эффективному в данном случае методу. Такая ситуация обычно возникает в окрестности точки минимума x^* или при «попадании в овраг».

Замечание 2′. Неравенство (2.1) эквивалентно неравенству

$$f_0(x^k) - f_0(x^{k+1}) \geqslant \lambda_k (f_0(x^k) - \beta_k).$$

Отсюда видно, что при малых λ_k точность вычислений β_k (как минимума функции f_0 по направлению $-\nabla f_0(x^k)$) может быть невысокой. Однако с уменьшением λ_k скорость сходимости уменьшается и поэтому важно удачно выбирать числа λ_k для каждого конкретного случая.

3. Основной вариант градиентного метода

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, произвольную константу $\rho > 0$, произвольный множитель $\beta \in [^1/_2, 1)$, произвольную константу $\varepsilon > 0$ ($\varepsilon \approx {}^1/_2$); положить $k = 0$.

Основной цикл. II. Вычислить $\nabla f_0(x^k)$.

III. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Положить $\alpha = \rho$.

V. Вычислить точку

$$x = x^k - \alpha \nabla f_0(x^k).$$

VI. Если выполняется неравенство

$$f_0(x) - f_0(x^k) \leqslant \varepsilon\alpha(\nabla f_0(x^k), \quad \nabla f_0(x^k)),$$

то положить $\rho_k = \alpha$ и перейти к шагу VII; иначе положить $\alpha = \alpha\beta$ и перейти к шагу V.

VII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \nabla f_0(x^k).$$

VIII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 3. *Если выполнено предположение 1 и (i) — функция f_0 ограничена снизу*

$$f_0(x) \geqslant \delta_0 > -\infty, \quad \forall x \in \boldsymbol{R}^n;$$

(ii) — градиент функции f_0 удовлетворяет условию Липшица

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \alpha_1 \|x - y\|, \quad \forall x, \ y \in \boldsymbol{R}^n, \quad \alpha_1 < \infty,$$

то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, справедливо

$$\|\nabla f_0(x^k)\| \to 0 \quad \text{при} \quad k \to \infty.$$

Теорема 3'. *Если функция f_0 дважды непрерывно дифференцируема в $\boldsymbol{R}^n$ и ее матрица вторых производных $\nabla^2_{xx} f_0(x)$ удовлетворяет неравенствам*

$$\gamma_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x) y, \ y) \leqslant \gamma_2 \|y\|^2;$$
$$\gamma_2 \geqslant \gamma_1 > 0$$

при любых $x, y \in R^n$, то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, справедливы соотношения

$$x^k \to x^* \quad \text{при} \quad k \to \infty;$$

$$f_0(x^k) \to f_0(x^*) \quad \text{при} \quad k \to \infty,$$

где x^* — единственная точка минимума функции f_0, и следующие оценки скорости сходимости

$$\|x^k - x^*\| \leqslant \delta_1 q^{k/2}, \quad \delta_1 < \infty;$$

$$f_0(x^k) - f_0(x^*) \leqslant (f_0(x^0) - f_0(x^*))\, q^k,$$

где $q = 1 - 2\varepsilon\,(1 - \varepsilon)\,\gamma_1\,(1 + \gamma_1/\gamma_2)/\gamma_2$, причем минимальное значение $q = 1 - (\gamma_1/2\gamma_2)(1 + \gamma_1/\gamma_2)$ достигается при $\varepsilon = {}^1/_2$.

Теорема 3″. *Если выполнены все условия теоремы 3 и существует такое число $\delta_2 > 0$, что*

$$\|\nabla f_0(x)\|^2 \geqslant \delta_2\,(f_0(x) - f_0^*), \quad \forall x \in R^n,$$

где $f_0^* = \inf\limits_{x \in R^n} f_0(x)$,

то для последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, справедлива оценка

$$f_0(x^k) - f_0^* \leqslant q_1^k\,(f_0(x^0) - f_0^*),$$

где $0 < q_1 < 1$.

4. Градиентный метод с постоянным шаговым множителем

Предположения 4. (*i*) — функция f_0 непрерывно дифференцируема; (*ii*) — градиент функции f_0 удовлетворяет условию Липшица с известной константой $\alpha < \infty$.

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$ и произвольную константу $\varepsilon \in (0, 1)$ (целесообразно выбирать $\varepsilon = {}^1/_2$); положить $k = 0$.

II. Выбрать произвольное значение шагового множителя ρ из полуоткрытого интервала $(0, (1 - \varepsilon)/\alpha]$.

Основной цикл. III. Вычислить $\nabla f_0(x^k)$.

IV. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу V.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho \nabla f_0(x^k).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 4. *Если выполнены предположения 4 и функция f_0 ограничена снизу, то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 4, справедливо $\|\nabla f_0(x^k)\| \to 0$ при $k \to \infty$.*

Предположения 4'. (*i*) — функция f_0 дважды непрерывно дифференцируема; (*ii*) — матрица вторых производных $\nabla^2_{xx} f_0(x)$ функции f_0 удовлетворяет неравенствам

$$\gamma_1 \| y \|^2 \leqslant (\nabla^2_{xx} f_0(x) y, \ y) \leqslant \gamma_2 \| y \|^2, \quad \gamma_2 \geqslant \gamma_1 > 0,$$

при любых $x, y \in R^n$, причем константы γ_2 и γ_1 известны.

Алгоритм 4'

Шаги I, III, IV, V, VI такие же, как и в алгоритме 4. Шаг II алгоритма 4 следует заменить шагом II', т. е. выбрать произвольное значение шагового множителя ρ из полуоткрытого интервала $(0, \ 2(1 - \varepsilon)/\gamma_2]$.

Теорема 4'. *Если выполнены предположения 4', то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 4', справедливы предельные соотношения*

$$\lim_{k \to \infty} x^k = x^*; \quad \lim_{k \to \infty} f_0(x^k) = f_0(x^*),$$

где x^* — единственная точка минимума функции f_0, и следующие оценки скорости сходимости:

$$\| x^k - x^* \| \leqslant \delta_1 q^{k/2}, \quad \delta_1 < \infty;$$

$$f_0(x^k) - f_0(x^*) \leqslant (f_0(x^0) - f_0(x^*)) q^k,$$

где $q = 1 - 2\varepsilon(1 - \varepsilon)\gamma_1(1 - \gamma_1/\gamma_2)/\gamma_2$, причем минимальное значение $q = 1 - (\gamma_1/2\gamma_2)(1 + \gamma_1/\gamma_2)$ достигается при $\varepsilon = 1/2$.

Алгоритм 4''

Шаги I, III, IV, V, VI такие, как и в алгоритме 4. Шаг II алгоритма 4 следует заменить шагом II'', т. е. выбрать произвольное значение шагового множителя ρ из открытого интервала $(0, \ 2/\gamma_2)$.

Теорема 4''. *Если выполнены предположения 4', то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 4'', справедлива оценка скорости сходимости*

$$\| x^k - x^* \| \leqslant q^k \| x^0 - x^* \|,$$

где $q = \max \{ | 1 - \rho\gamma_1 |, \ | 1 - \rho\gamma_2 | \}$, причем минимальное значение $q = (\gamma_2 - \gamma_1)/(\gamma_2 + \gamma_1)$ достигается при $\rho = 2/(\gamma_2 + \gamma_1)$.

Если же в алгоритме 4'' выбрать шаговый множитель $\rho = 2/(\gamma_2 + \gamma_1)$, то, кроме того, справедливо неравенство

$$f_0(x^{k+1}) - f_0(x^*) \leqslant ((\gamma_2 - \gamma_1)/(\gamma_2 + \gamma_1))^2 (f_0(x^k) - f_0(x^*)).$$

5. Вариант градиентного метода с матрицей ускорения сходимости

Предположения 5. (*i*) — функция f_0 непрерывно дифференцируема; (*ii*) — задана матрица ускорения сходимости $H(x)$ размера $n \times n$, элементами которой являются непрерывные функции от x.

Алгоритм 5

Начало. I. Выбрать произвольное начальное приближение

$x^0 \in R^n$ и произвольные константы $\varepsilon \in (0, 1)$, $\beta \in (0, 1)$, $\rho > 0$ (рекомендуется выбрать $\varepsilon = 1/2$, $\beta \in (1/2, 4/5)$, $\rho = 1$); положить $k = 0$.

Основной цикл. II. Вычислить вектор движения $h(x^k)$ к следующему приближению x^{k+1} по формуле

$$h(x^k) = -H(x^k)\nabla f_0(x^k).$$

III. Если $h(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Положить $\mu = \rho$.

V. Вычислить значение

$$\theta(\mu, x^k) = f_0(x^k + \mu h(x^k)) - f_0(x^k) - \mu\varepsilon(\nabla f_0(x^k), h(x^k)).$$

VI. Если $\theta(\mu, x^k) \leqslant 0$, то положить $\rho_k = \mu$ и перейти к шагу VII; иначе положить $\mu = \beta\mu$ и перейти к шагу V.

VII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h(x^k).$$

VIII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 5. *Если выполнены предположения 5 и условия: (i) — функция f_0 дважды непрерывно дифференцируема; (ii) — матрица вторых производных $\nabla^2_{xx} f_0(x)$ функции f_0 удовлетворяет неравенствам*

$$\gamma_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x)\, y,\, y) \leqslant \gamma_2 \|y\|^2, \quad \gamma_2 \geqslant \gamma_1 > 0,$$

при любых x, $y \in R^n$; (iii) — $H(x)$ — симметричная матрица, удовлетворяющая неравенствам

$$\gamma_3 \|y\|^2 \leqslant (H(x)\, y,\, y) \leqslant \gamma_4 \|y\|^2, \quad \gamma_4 \geqslant \gamma_3 > 0,$$

при любых x, $y \in R^n$, то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 5, справедливы оценки скорости сходимости

$$\|x^k - x^*\| \leqslant \delta_1 q^{k/2}, \quad \delta_1 < \infty;$$

$$f_0(x^k) - f_0(x^*) \leqslant (f_0(x^0) - f_0(x^*))\, q^k,$$

где $q = 1 - 2\varepsilon(1 - \varepsilon)\gamma_1\gamma_3(1 + \gamma_1/\gamma_2)/(\gamma_2\gamma_4)$; x^* — единственная точка минимума функции f_0, причем минимальное значение $q = 1 - \gamma_1\gamma_3(1 + \gamma_1/\gamma_2)/(2\gamma_2\gamma_4)$ достигается при $\varepsilon = 1/2$.

Теорема 5′. *Если выполнены предположения 5 и условия: (i) — $H(x)$ — положительно-определенная матрица; (ii) — начальная точка x^0 такова, что множество $X_0 = \{x \mid f_0(x) \leqslant f_0(x^0),\ x \in R^n\}$ ограничено, то каждая предельная точка x' бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 5, удовлетворяет условию $\nabla f_0(x') = 0$.*

Замечание 5. Выбор матрицы $H(x)$ существенно влияет на скорость сходимости алгоритма 5 (примером этого служат методы типа Ньютона и методы с растяжением пространства).

6. Модифицированный градиентный метод, не требующий вычисления производных

Этот метод применяется в тех случаях, когда вычисление градиента $\nabla_x f_0$ требует значительно больших затрат, чем вычисление значений минимизируемой функции f_0, или когда вычисление градиента вообще невозможно (например, когда функция f_0 задана таблично, или вычисляется с помощью экспериментов).

Алгоритм 6

Начало. I. Выбрать начальное приближение $x^0 \in \boldsymbol{R}^n$, удовлетворяющее условиям теоремы 6, и константы $\varepsilon_0 > 0$, $\lambda' > 0$, $\beta > 0$, $\lambda \in (0, 1/2)$ (рекомендуется выбирать $\varepsilon_0 \in [10^{-3}, 10^{-2}]$, $\lambda' \in [10^{-3}, 10^{-2}]$, $\beta \in [5, 10]$, $\lambda = 0{,}4$).

II. Положить $k = 0$, $\varepsilon = \varepsilon_0$.

Основной цикл. III. Вычислить вектор h^k, j-я компонента h_j^k которого

$$h_j^k = -\frac{1}{\varepsilon}(f_0(x^k + \varepsilon e^j) - f_0(x^k)), \quad j = 1, 2, \ldots, n,$$

где e^j — j-й орт.

IV. Вычислить

$$\Delta_k = f_0(x^k + \varepsilon\beta h^k) - f_0(x^k).$$

V. Если $\Delta_k < 0$, то, используя алгоритм 6А, вычислить такое число $\bar{\rho}$, что

$$\frac{\bar{\rho}(1-\lambda)\Delta_k}{\beta\varepsilon} \leqslant f_0(x^k + \bar{\rho}h^k) - f_0(x^k) \leqslant \frac{\bar{\rho}\lambda\Delta_k}{\beta\varepsilon}$$

и перейти к шагу VI; иначе положить $\varepsilon = \varepsilon/2$ и перейти к шагу III.

VI. Если $f_0(x^k + \bar{\rho}h^k) - f_0(x^k) \leqslant -\lambda'\varepsilon$, то перейти к шагу VII; иначе положить $\varepsilon = \varepsilon/2$ и перейти к шагу III.

VII. Вычислить следующее приближение

$$x^{k+1} = x^k + \bar{\rho}h^k.$$

VIII. Положить $k = k + 1$ и перейти к шагу III.

Алгоритм 6А (алгоритм вычисления шагового множителя $\bar{\rho}$ для алгоритма 6)

I. Выбрать константу $\rho > 0$ (рекомендуется выбирать $\rho = 1$).

II. Определить функции

$$\psi_k(\alpha) = f_0(x^k + \alpha h^k) - f_0(x^k) - \frac{\alpha(1-\lambda)\Delta_k}{\beta\varepsilon};$$

$$\varphi_k(\alpha) = f_0(x^k + \alpha h^k) - f_0(x^k) - \frac{\alpha\lambda\Delta_k}{\beta\varepsilon}.$$

III. Положить $\mu = \rho$.

IV. Вычислить значение $\psi_k(\mu)$.

V. Если $\psi_k(\mu) = 0$, то положить $\bar{\rho} = \mu$ и прекратить вычисления; если $\psi_k(\mu) < 0$, то положить $\mu = \mu + \rho$ и перейти к шагу IV; если $\psi_k(\mu) > 0$, то перейти к шагу VI.

VI. Вычислить значение $\varphi_k(\mu)$.

VII. Если $\varphi_k(\mu) \leqslant 0$, то положить $\bar{\rho} = \mu$ и прекратить вычисления; иначе положить $a_0 = \mu - \rho$, $b_0 = \mu$ и перейти к шагу VIII.

VIII. Положить $i = 0$.

IX. Вычислить $v_i = (a_i + b_i)/2$.

X. Вычислить значения $\psi_k(v_i)$ и $\varphi_k(v_i)$.

XI. Если $\psi_k(v_i) \geqslant 0$ и $\varphi_k(v_i) \leqslant 0$, то положить $\bar{\rho} = v_i$ и прекратить вычисления; иначе перейти к шагу XII.

XII. Если $\psi_k(v_i) > 0$, то положить $a_{i+1} = a_i$, $b_{i+1} = v_i$, $i = i + 1$ и перейти к шагу IX; иначе положить $a_{i+1} = v_i$, $b_{i+1} = b_i$, $i = i + 1$ и перейти к шагу IX.

Теорема 6. *Если выполнены условия: (i) — функция f_0 непрерывно дифференцируема; (ii) — начальное приближение x^0 в алгоритме 6 таково, что множество $X_0 = \{x \mid f_0(x) \leqslant f_0(x^0),\ x \in \mathbf{R}^n\}$ ограничено, то последовательность $x^0, x^1, \ldots$, порожденная алгоритмом 6, либо конечна (т. е. после конечного числа итераций алгоритм 6 зациклится в точке x^k между шагами III и V или III и VI, повторяя деление ε на 2), причем $\nabla f_0(x^k) = 0$, либо бесконечна и каждая ее предельная точка x' удовлетворяет условию $\nabla f_0(x') = 0$.*

Алгоритм 6'

Все шаги алгоритма 6, за исключением шага V, остаются без изменений. Шаг V алгоритма 6 следует заменить шагом V', т. е. если $\Delta_k < 0$, то, используя алгоритм 6Б, вычислить такое число $\bar{\rho}$, что

$$f_0(x^k + \bar{\rho}h^k) - f_0(x^k) \leqslant \frac{\bar{\rho}\lambda\Delta_k}{\beta\varepsilon}$$

и перейти к шагу VI; иначе положить $\varepsilon = \varepsilon/2$ и перейти к шагу III.

Алгоритм 6Б (алгоритм вычисления шагового множителя $\bar{\rho}$ для алгоритма 6').

I. Выбрать константы $\gamma' \in (0, 1)$, $\rho > 0$ (рекомендуется выбирать $\gamma' \in (1/2, 4/5)$, $\rho = 1$).

II. Определить функцию $\varphi_k : \mathbf{R}^1 \to \mathbf{R}^1$ равенством

$$\varphi_k(\alpha) = f_0(x^k + \alpha h^k) - f_0(x^k) - \frac{\alpha\lambda\Delta_k}{\beta\varepsilon}.$$

III. Положить $\mu = \rho$.

IV. Вычислить значение $\varphi_k(\mu)$.

V. Если $\varphi_k(\mu) \leqslant 0$, то положить $\bar{\rho} = \mu$ и прекратить вычисления, иначе положить $\mu = \mu\gamma'$ и перейти к шагу IV.

Для алгоритма 6' справедлива теорема, аналогичная теореме 6.

Библиографические указания. При написании параграфа использовались работы [320, 194, 285]. Многие результаты получены ранее [446, 185, 278, 536, 422, 486, 453, 198, 286, 421; 225, 390, 104, 49, 48, 376, 397, 439, 410].

2.2. Методы типа Ньютона

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной непрерывно дифференцируемой функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$.

1. Метод Ньютона—Канторовича

Предположение 1. Минимизируемая функция f_0 дважды непрерывно дифференцируема с обратимой матрицей Гессе $\nabla^2_{xx} f_0(x)$.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, положить $k = 0$.

Основной цикл. II. Вычислить вектор движения h^k к следующему приближению x^{k+1} из системы уравнений

$$-\nabla^2_{xx} f_0(x^k)\, h^k = \nabla f_0(x^k).$$

III. Если $h^k = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Вычислить следующее приближение

$$x^{k+1} = x^k + h^k.$$

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Пусть выполнено предположение 1. Тогда, если бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к точке x', то $\nabla f_0(x') = 0$.*

2. Обобщенный метод Ньютона—Канторовича

Предположение 2. (*i*) — функция f_0 дважды непрерывно дифференцируема с обратимой матрицей Гессе $\nabla^2_{xx} f_0(x)$, удовлетворяющей условию

$$\gamma_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x)\, y,\ y) \leqslant \gamma_2 \|y\|^2, \quad \gamma_2 \geqslant \gamma_1 > 0.$$

(Отметим, что у таких функций существует единственная точка минимума x^*).

Алгоритм 2

Шаги I—III такие, как в алгоритме 1.

IV. Вычислить шаговый множитель ρ_k, удовлетворяющий условию

$$f_0(x^k + \rho_k h^k) = \min_{\rho \geqslant 0} f_0(x^k + \rho h^k).$$

V. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

VI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. Если выполнены предположения 2, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, сходится к решению x^* со сверхлинейной скоростью, т. е.

$$\| x^{k+1} - x^* \| \leqslant \nu_k \| x^k - x^* \|,$$

где $\quad \nu_k = (\gamma_2/\gamma_1^3)^{1/2} \| \nabla^2_{xx} f_0(x^k) - \nabla^2_{xx} f_0(x^k - \theta(x^k - x^*)) \|, \quad \theta \in [0,\ 1].$

Если, кроме того, матрица вторых производных $\nabla^2_{xx} f_0(x)$ функции f_0 удовлетворяет условию

$$\| \nabla^2_{xx} f_0(x) - \nabla^2_{xx} f_0(y) \| \leqslant \alpha \| x - y \|, \quad \alpha < \infty,$$

при любых $x, y \in \boldsymbol{R}^n$, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$ сходится к решению x^* с квадратичной скоростью, т. е.

$$\| x^{k+1} - x^* \| \leqslant (\gamma_2/\gamma_1)^{1/2} (\alpha/\gamma_1) \| x^k - x^* \|^2.$$

Такой вариант обобщенного метода Ньютона — Канторовича в общем случае реализовать на цифровой ЭВМ невозможно из-за требования одномерной минимизации на шаге IV алгоритма 2. Приведенный ниже вариант свободен от этого недостатка.

Алгоритм 2'

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$ и произвольные константы $\varepsilon \in (0,\ ^1/_2)$, $\beta \in (0,\ 1)$, положить $\rho = 1$ и $k = 0$.

Основной цикл. II. Вычислить вектор движения h^k к следующему приближению x^{k+1}

$$h^k = -(\nabla^2_{xx} f_0(x^k))^{-1} \nabla f_0(x^k).$$

III. Если $h^k = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Положить $\mu = \rho$.

V. Вычислить значение

$$\delta = f_0(x^k + \mu h^k) - f_0(x^k) - \varepsilon\mu(\nabla f_0(x^k),\ h^k).$$

VI. Если $\delta \leqslant 0$, то положить $\rho_k = \mu$ или вычислить ρ_k из неравенств

$$\rho_k(1 - \varepsilon)(\nabla f_0(x^k),\ h^k) \leqslant f_0(x^k + \rho_k h^k) - f_0(x^k) \leqslant \rho_k \varepsilon (\nabla f_0(x^k),\ h^k)$$

и перейти к шагу VII; иначе положить $\mu = \mu\beta$ и перейти к шагу V.

VII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

VIII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2'. Если выполнены предположения 2, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2', сходится к точке минимума x^* со сверхлинейной скоростью, т. е.

$$\| x^{N+l} - x^* \| \leqslant \eta \lambda_N \ldots \lambda_{N+l},$$

где N, $\eta < \infty$; $\lambda_{N+l} < 1$ при любом $l \geqslant 0$; $\lambda_i \to 0$ при $i \to \infty$.

Если, кроме того, матрица вторых производных $\nabla^2_{xx} f_0(x)$ *функции* f_0 *удовлетворяет условию*

$$\| \nabla^2_{xx} f_0(x) - \nabla^2_{xx} f_0(y) \| \leqslant \alpha \| x - y \|, \quad \alpha < \infty,$$

при любых $x,\ y \in \boldsymbol{R}^n$, *то бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$ *сходится к решению* x^* *с квадратичной скоростью, т. е.*

$$\| x^{k+1} - x^* \| \leqslant (\alpha/\gamma_1) \| x^k - x^* \|^2.$$

Замечание 2. Используемый в алгоритме 2 способ выбора значения шагового множителя ρ_k, $k = 0, 1, \ldots$, при выполнении условий теоремы 2′ гарантирует, что, начиная с некоторой итерации, алгоритм 2′ будет осуществляться с единичным шаговым множителем $\rho_k = 1$, т. е. алгоритм 2′ вырождается в обычный метод Ньютона — Канторовича (алгоритм 1).

3. Модификации обобщенного метода Ньютона—Канторовича

Алгоритм 3

Алгоритм 3 отличается от алгоритма 2′ только более простым способом вычисления вектора h^k на шаге II из системы уравнений

$$-(\nabla^2_{xx} f_0(x^0))\, h^k = \nabla f_0(x^k).$$

Теорема 3. *Если выполнены предположения 2 и матрица* $H = (\nabla^2_{xx} f_0(x^0))^{-1}$ *удовлетворяет условию*

$$\gamma_3 \| y \|^2 \leqslant (Hy,\ y) \leqslant \gamma_4 \| y \|^2, \quad \gamma_4 \geqslant \gamma_3 > 0, \quad \forall y \in \boldsymbol{R}^n,$$

то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом* 3, *справедливы оценки скорости сходимости*

$$\| x^k - x^* \| \leqslant \delta_1 \bar{q}^{k/2}, \quad \delta_1 < \infty;$$

$$f_0(x^k) - f_0(x^*) \leqslant (f_0(x^0) - f_0(x^*))\, q^k,$$

где $q = 1 - 2\varepsilon(1 - \varepsilon)\, \gamma_1 \gamma_3 (1 + \gamma_1/\gamma_2)/(\gamma_2 \gamma_4)$.

Если, кроме того, начальное приближение x^0 *в алгоритме* 3 *выбрано достаточно близко к точке* x^*, *то* $\rho_k = 1$, $k = 0, 1, \ldots$, *и последовательность* $\{x^k\}_{k=0}^{\infty}$ *сходится к* x^* *со сверхлинейной скоростью сходимости*

$$\| x^{k+1} - x^* \| \leqslant q_1 \| x^k - x^* \|,$$

где

$$q_1 = (1/\gamma_1) \max_{x \in X_0} \| \nabla^2_{xx} f_0(x^0) - \nabla^2_{xx} f_0(x) \|,$$

$$X_0 = \{ x \mid f_0(x) \leqslant f_0(x^0), \quad x \in \boldsymbol{R}^n \}.$$

В алгоритме 3 на каждой итерации для построения вектора движения h^k используется одна и та же матрица — $\nabla^2_{xx} f_0(x^0)$. Ниже приводится модификация обобщенного метода Ньютона — Канторовича, в которой обновление матрицы производится через τ $(\tau \geqslant 1)$

шагов. Такой алгоритм занимает промежуточное значение между алгоритмом 3 и обобщенным методом Ньютона — Канторовича (алгоритм 2).

Алгоритм 3'

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, константы $\varepsilon \in (0, 1/2)$, $\beta \in (0, 1)$ и натуральное число $\tau \geqslant 1$.

II. Положить $j = 0$, $\rho = 1$.

Основной цикл. III. Положить $i = 0$.

IV. Вычислить индекс $k = j\tau + i$.

V. Вычислить вектор движения h^k к следующему приближению x^{k+1} из системы

$$-\nabla^2_{xx} f_0(x^{j\tau})\, h^k = \nabla f_0(x^k).$$

VI. Если $h^k = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VII.

VII. Положить $\mu = \rho$.

VIII. Вычислить значение

$$\delta = f_0(x^k + \mu h^k) - f_0(x^k) - \varepsilon\mu\,(\nabla f_0(x^k),\ h^k).$$

IX. Если $\delta \leqslant 0$, то положить $\rho_k = \mu$ и перейти к шагу X; иначе положить $\mu = \mu\beta$ и перейти к шагу VIII.

X. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

XI. Если $i < \tau - 1$, то положить $i = i + 1$ и перейти к шагу IV; иначе перейти к шагу XII.

XII. Положить $j = j + 1$ и перейти к шагу III.

Теорема 3'. *Если выполнены предположения* 2 *и*

$$\|\nabla^2_{xx} f_0(x) - \nabla^2_{xx} f_0(y)\| \leqslant \alpha \|x - y\|, \quad \alpha < \infty, \quad \forall\, x,\ y \in R^n,$$

то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом* 3', *сходится к решению* x^* *со сверхлинейной скоростью*

$$\|x^{(j+1)\tau} - x^*\| \leqslant \nu \|x^{j\tau} - x^*\|^{\tau+1}.$$

Замечание 3. Сходимость с любого начального приближения является существенным преимуществом обобщенного метода Ньютона — Канторовича и его модификаций по сравнению с обычным методом Ньютона — Канторовича, в котором сходимость гарантируется лишь при наличии достаточно хорошего начального приближения.

4. Модифицированный обобщенный метод Ньютона—Канторовича, не требующий вычисления матрицы вторых производных

Алгоритм 4

Начало. I. Выбрать начальное приближение $x^0 \in R^n$, удовлетворяющее условиям теоремы 4, и константы $\varepsilon_0 > 0$, $\beta > 0$,

$\rho > 0$, $\alpha \in (0, 1/2)$ (рекомендуется выбирать: $\varepsilon_0 \in [10^{-3}, 10^{-2}]$, $\beta \in [10^{-3}, 10^{-2}]$, $\alpha = 0{,}4$; $\rho = 1$).

II. Положить $k = 0$.

Основной цикл. III. Вычислить $\nabla f_0 (x^k)$.

IV. Если $\nabla f_0 (x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу V.

V. Найти ε из условия $\varepsilon = \min \{\varepsilon_0, \| \nabla f_0 (x^k) \|\}$.

VI. Найти матрицу $H_k (\varepsilon)$ размера $n \times n$, j-й столбец которой вычисляется по формуле

$$\frac{1}{\varepsilon} (\nabla f_0 (x^k + \varepsilon e^j) - \nabla f_0 (x^k)), \quad j = 1, 2, \ldots, n,$$

где e^j — j-й орт.

VII. Если существует обратная матрица $H_k^{-1} (\varepsilon)$ и выполняется неравенство $(\nabla f_0 (x^k), H_k^{-1} (\varepsilon) \nabla f_0 (x^k)) > 0$, то вычислить вектор

$$h^k (\varepsilon) = - H_k^{-1} (\varepsilon) \nabla f_0 (x^k)$$

и перейти к шагу VIII; иначе положить $\varepsilon = \varepsilon/2$ и перейти к шагу VI.

(Шаги VIII—XVIII предназначены для вычисления такого $\bar{\rho}$, что

$$\bar{\rho} (1 - \alpha) (\nabla f_0 (x^k), h^k (\varepsilon)) \leqslant f_0 (x^k + \bar{\rho} h^k (\varepsilon)) - f_0 (x^k) \leqslant \bar{\rho} \alpha (\nabla f_0 (x^k), h^k (\varepsilon)).$$

VIII. Определить функции

$$\theta_k (\lambda) = f_0 (x^k + \lambda h^k (\varepsilon)) - f_0 (x^k) - \lambda (1 - \alpha) (\nabla f_0 (x^k), h^k (\varepsilon));$$
$$\varphi_k (\lambda) = f_0 (x^k + \lambda h^k (\varepsilon)) - f_0 (x^k) - \lambda \alpha (\nabla f_0 (x^k), h^k (\varepsilon)).$$

IX. Положить $\mu = \rho$.

X. Вычислить $\theta_k (\mu)$.

XI. Если $\theta_k (\mu) = 0$, то положить $\bar{\rho} = \mu$ и перейти к шагу XIX; если $\theta_k (\mu) < 0$, то положить $\mu = \mu + \rho$ и перейти к шагу X; если $\theta_k (\mu) > 0$, то перейти к шагу XII.

XII. Вычислить $\varphi_k (\mu)$.

XIII. Если $\varphi_k (\mu) \leqslant 0$, то положить $\bar{\rho} = \mu$ и перейти к шагу XIX; иначе положить $a_0 = \mu - \rho$, $b_0 = \mu$ и перейти к шагу XIV;

XIV. Положить $i = 0$.

XV. Вычислить значение $v_i = (a_i + b_i)/2$.

XVI. Вычислить $\theta_k (v_i)$, $\varphi_k (v_i)$.

XVII. Если $\theta_k (v_i) \geqslant 0$ и $\varphi_k (v_i) \leqslant 0$, то положить $\bar{\rho} = v_i$ и перейти к шагу XIX; иначе перейти к шагу XVIII.

XVIII. Если $\theta_k (v_i) > 0$, то положить $a_{i+1} = a_i$, $b_{i+1} = v_i$, $i = i + 1$ и перейти к шагу XV; иначе положить $a_{i+1} = v_i$, $b_{i+1} = b_i$, $i = i + 1$ и перейти к шагу XV.

XIX. Если выполняется неравенство

$$f_0 (x^k + \bar{\rho} h^k (\varepsilon)) - f_0 (x^k) \leqslant - \beta \varepsilon,$$

то перейти к шагу XX; иначе положить $\varepsilon = \varepsilon/2$ и перейти к шагу VI.

XX. Вычислить следующее приближение

$$x^{k+1} = x^k + \bar{\rho} h^k(\varepsilon).$$

XXI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 4. *Если выполнены условия:* (*i*) *— функция* f_0 *дважды непрерывно дифференцируема и ее матрица Гессе невырождена при всех* x *из достаточно большой области;* (*ii*) *— функция* f_0 *строго выпукла;* (*iii*) *— начальное приближение* x^0 *в алгоритме 4 таково, что множество* $X_0 = \{x \mid f_0(x) \leqslant f_0(x^0),\ x \in \boldsymbol{R}^n\}$ *ограничено, то бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 4, сходится к точке минимума* x^* *функции* f_0.

В работе [285] утверждается, что алгоритм 4 имеет сверхлинейную сходимость.

Библиографические указания. Параграф основан на работах [285, 320, 486]. Методы Ньютона решения задач безусловной оптимизации изучались также [186, 16, 49, 413, 423, 200, 105, 106, 278, 62, 498, 390].

В работах [469, 544] изучается метод Ньютона для вырожденных случаев. В работах [537, 538] приводятся модели обобщенных алгоритмов и на их основании устанавливается глобальная сходимость модификаций метода Ньютона.

2.3. Методы двойственных направлений

Задача 0. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной непрерывно дифференцируемой функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$.

Методы двойственных направлений, обладая сверхлинейной скоростью сходимости, требуют, по сравнению с методами типа Ньютона — Канторовича, значительно меньшего количества вычислений (примерно в n раз), при этом нет необходимости вычислять вторые производные функции f_0. С помощью лишь первых производных так организуется процесс вычислений, что направление поиска приближается к направлению метода Ньютона — Канторовича.

1. Основной вариант

Алгоритм 1

Начало. I. Выбрать: произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$; произвольную линейно-независимую систему n-мерных векторов $p^{0,0}, p^{0,-1}, \ldots, p^{0,-(n-1)}$ (в частности, можно взять столбцы единичной матрицы I размера $n \times n$); константы $\gamma > 0$ (рекомендуется выбирать $\gamma \in [10^{-15}, 10^{-5}]$), $\beta \in (0,1)$ (рекомендуется выбирать $\beta = 0{,}8$), $\varepsilon \in (0, {}^1/_2)$.

II. Положить $k = 0$, $\rho = 1$, $r^0 = p^{0,0}$.

Основной цикл. III. Вычислить $\nabla f_0(x^k)$.

IV. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу V.

V. Положить $\rho_k = \rho$.

VI. Вычислить точку $x = x^k - \rho_k \nabla f_0(x^k)$.

VII. Если выполняется неравенство

$$f_0(x) - f_0(x^k) \leqslant -\varepsilon\rho_k (\nabla f_0(x^k), \nabla f_0(x^k)),$$

то перейти к шагу VIII; иначе положить $\rho_k = \rho_k\beta$ и перейти к шагу VI.

VIII. Положить $x^{k+1} = x$ и перейти к шагу IX.

IX. Вычислить векторы

$$r^{k+1} = x^{k+1} - x^k;$$

$$g^{k+1} = \nabla f_0(x^{k+1}) - \nabla f_0(x^k).$$

X. Если выполняется неравенство

$$(p^{k,k-(n-1)}, g^{k+1}) \geqslant \gamma \| p^{k,k-(n-1)} \| \cdot \| g^{k+1} \|,$$

то перейти к шагу XV; иначе положить $\delta = 1$ и перейти к шагу XI.

XI. Вычислить вектор $r^{k+1} = \delta p^{k\,k-(n-1)}$.

XII. Если $\| r^{k+1} \| < \| r^k \|$, то перейти к шагу XIII; иначе положить $\delta = \delta\beta$ и перейти к шагу XI.

XIII. Вычислить $\nabla f_0(x^k + r^{k+1})$.

XIV. Вычислить вектор

$$g^{k+1} = \nabla f_0(x^k + r^{k+1}) - \nabla f_0(x^k).$$

XV. Вычислить вектор

$$p^{k+1,k+1} = (p^{k,k-(n-1)}, g^{k+1})\, p^{k,k-(n-1)}.$$

XVI. Положить $j = 0$.

XVII. Вычислить вектор

$$p^{k+1,k-j} = p^{k,k-j} - (p^{k,k-j}, g^{k+1})\, p^{k+1,k+1}.$$

XVIII. Если $j < n - 2$, то положить $j = j + 1$ и перейти к шагу XVII; иначе перейти к шагу XIX.

XIX. Если $k < n - 1$, то положить $k = k + 1$ и перейти к шагу III; иначе перейти к шагу XX.

XX. Вычислить $\nabla f_0(x^{k+1})$.

XXI. Если $\nabla f_0(x^{k+1}) = 0$, то положить $x^* = x^{k+1}$ и прекратить вычисления; иначе перейти к шагу XXII.

XXII. Вычислить вектор

$$h^{k+1} = -\sum_{i=0}^{n-1} (\nabla f_0(x^{k+1}), p^{k+1,k+1-i})\, r^{k+1-i}.$$

XXIII. Если $(\nabla f_0(x^{k+1}), h^{k+1}) < 0$, то положить $\rho_{k+1} = \rho$ и перейти к шагу XXIV; иначе перейти к шагу XXVI.

XXIV. Вычислить точку $x = x^{k+1} + \rho_{k+1} h^{k+1}$.

XXV. Если выполняется неравенство

$$f_0(x) - f_0(x^{k+1}) \leqslant \varepsilon\rho_{k+1} (\nabla f_0(x^{k+1}), h^{k+1}),$$

то положить $x^{k+2} = x$ и перейти к шагу XXXI; иначе положить $\rho_{k+1} = \beta\rho_{k+1}$ и перейти к шагу XXIV.

XXVI. Если $(\nabla f_0\,(x^{k+1}),\ h^{k+1}) > 0$, то положить $\rho_{k+1} = \rho$ и перейти к шагу XXVII; иначе положить $\rho_{k+1} = \rho$ и перейти к шагу XXIX.

XXVII Вычислить точку $x = x^{k+1} - \rho_{k+1}h^{k+1}$.

XXVIII. Если выполняется неравенство

$$f_0(x) - f_0(x^{k+1}) \leqslant -\varepsilon\rho_{k+1}(\nabla f_0(x^{k+1}),\ h^{k+1}),$$

то положить $x^{k+2} = x$ и перейти к шагу XXXI; иначе положить $\rho_{k+1} = \beta\rho_{k+1}$ и перейти к шагу XXVII.

XXIX. Вычислить точку $x = x^{k+1} - \rho_{k+1}\nabla f_0\,(x^{k+1})$.

XXX. Если выполняется неравенство

$$f_0(x) - f_0(x^{k+1}) \leqslant \varepsilon\rho_{k+1}\,\|\nabla f_0(x^{k+1})\|^2,$$

то положить $x^{k+2} = x$ и перейти к шагу XXXI; иначе положить $\rho_{k+1} = \beta\rho_{k+1}$ и перейти к шагу XXIX.

XXXI. Положить $k = k + 1$ и перейти к шагу IX.

Теорема 1. *Если функция f_0 дважды непрерывно дифференцируема и ее матрица вторых производных удовлетворяет условию*

$$\gamma_1\|y\|^2 \leqslant (\nabla^2_{xx}f_0(x)\,y,\ y) \leqslant \gamma_2\|y\|^2,\quad \gamma_2 \geqslant \gamma_1 > 0$$

при любых $x,\ y \in R^n$, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, *сходится к решению x^* задачи 0 со сверхлинейной скоростью*

$$\|x^{N+l} - x^*\| \leqslant \alpha\lambda_N\lambda_{N+1}\ \ldots\ \lambda_{N+l},$$

где α, $N < \infty$; $\lambda_{N+l} < 1$ при любом $l \geqslant 0$; $\lambda_i \to 0$ при $i \to \infty$, причем $f_0\,(x^{k+1}) < f_0\,(x^k)$, $k = 0, 1, \ldots$.

Для реализации методов двойственных направлений на ЭВМ требуется хранить в памяти две системы векторов:

$$r^k,\ r^{k-1},\ \ldots,\ r^{k-(n-1)} \quad \text{и} \quad p^{k,k},\ p^{k,k-1},\ \ldots,\ p^{k,k-(n-1)},$$

т. е. фактически две матрицы размера $n \times n$. Частично этот недостаток устраняется, если в качестве векторов r^k использовать векторы, направленные по координатным осям, так как в этом случае в памяти машины необходимо хранить лишь один n-мерный вектор вместо системы $r^k, \ldots, r^{k-(n-1)}$.

2. Модифицированный метод двойственных направлений, не требующий вычисления производных

Алгоритм 2

Начало. I. Выбрать: произвольное начальное приближение $x^0 \in R^n$; произвольную систему n-мерных линейно независимых векторов $p^{0,0}, p^{0,-1}, \ldots, p^{0,-(n-1)}$ (в частности, можно взять столбцы единичной матрицы I размера $n \times n$); константы $\gamma_0 > 0$, $\alpha_0 \in (0, 1)$, $\varepsilon \in (0, {}^1/_2)$, $\delta > 0$, $t > 1$; последовательность $\{\xi_k\}_{k=0}^{\infty}$, стремя-

щуюся к нулю достаточно медленно (в частности, можно выбрать $\xi_k = 1/(k+1)$, $k = 0, 1, \ldots$).

II. Положить $k = 0$ и перейти к шагу XIV.

Основной цикл. III. Вычислить вектор r^k, используя для этого алгоритм 2А, или по формуле $r^k = x^k - x^{k-1}$.

IV. Вычислить вектор $y^k = x^k + r^k$.

V. Вычислить $\mu = \| r^k \|^t$.

VI. Вычислить n-мерные векторы $\bar\theta^k$, $\bar\varphi^k$, $\bar\psi^k$, j-е компоненты которых

$$\bar\theta_j^k = (f_0(x^k + \mu e^j) - f_0(x^k))/\mu, \quad j = 1, \ldots, n;$$

$$\bar\varphi_j^k = (f_0(y^k + \mu e^j) - f_0(y^k))/\mu, \quad j = 1, \ldots, n;$$

$$\bar\psi_j^k = \bar\varphi_j^k - \bar\theta_j^k, \quad j = 1, \ldots, n,$$

где e^j — j-й орт.

VII. Положить $\psi^k = \bar\psi^k$.

VIII. Вычислить скалярное произведение $(p^{k-1,k-n}, \psi^k)$.

IX. Если $(p^{k-1,k-n}, \psi^k) = 0$, то положить $\mu = \mu/2$ и перейти к шагу VI; иначе перейти к шагу X.

X. Вычислить вектор

$$p^{k,k} = (1/(p^{k-1,k-n}, \psi^k))\, p^{k-1,k-n}.$$

XI. Положить $i = 0$.

XII. Вычислить вектор

$$p^{k,k-1-i} = p^{k-1,k-1-i} - (p^{k-1,k-1-i}, \psi^k)\, p^{k,k}.$$

XIII. Если $i < n-2$, то положить $i = i+1$ и перейти к шагу XII; иначе перейти к шагу XIV.

XIV. Если $k \leqslant n-1$, то положить $\alpha = \alpha_0$ и перейти к шагу XV; если $k = n$, то положить $\alpha = \mu$ и перейти к шагу XXI; иначе перейти к шагу XIX.

XV. Вычислить вектор $\bar g^k$, j-я компонента которого

$$\bar g_j^k = (f_0(x^k + \alpha e^j) - f_0(x^k))/\alpha, \quad j = 1, \ldots, n.$$

XVI. Если $f_0(x^k - \bar g^k) < f_0(x^k)$, то положить $x^{k+1} = x^k - \bar g^k$ и перейти к шагу XVIII; иначе перейти к шагу XVII.

XVII. Если $f_0(x^k + \bar g^k) < f_0(x^k)$, то положить $x^{k+1} = x^k + \bar g^k$ и перейти к шагу XVIII; иначе положить $\alpha = \alpha/2$ и перейти к шагу XV.

XVIII. Положить $k = k+1$ и перейти к шагу III.

XIX. Вычислить $\| h^{k-1} \|^2$.

XX. Если $\| h^{k-1} \|^2 \leqslant \mu$, то положить $\alpha = \| h^{k-1} \|^2$ и перейти к шагу XXI; иначе положить $\alpha = \mu$ и перейти к шагу XXI.

XXI. Вычислить вектор $\bar g^k$, j-я компонента которого

$$\bar g_j^k = (f_0(x^k + \alpha e^j) - f_0(x^k))/\alpha, \quad j = 1, \ldots, n.$$

XXII. Положить $g^k = \bar{g}^k$.

XXIII. Вычислить вектор

$$\bar{h}^k = \sum_{i=0}^{n-1} (p^{k,k-i}, g^k) r^{k-i}.$$

XXIV. Если $\alpha \leqslant \xi_k \| \bar{h}^k \|$, то перейти к шагу XXV; иначе положить $\alpha = \alpha/2$ и перейти к шагу XXI.

XXV. Если выполняется неравенство $f_0(x^k - \gamma_0 \bar{h}^k) < f_0(x^k)$, то положить $h^k = -\bar{h}^k$ и перейти к шагу XXVIII; иначе перейти к шагу XXVI.

XXVI. Если выполняется неравенство $f_0(x^k + \gamma_0 \bar{h}^k) < f_0(x^k)$, то положить $h^k = \bar{h}^k$ и перейти к шагу XXVIII; иначе перейти к шагу XXVII.

XXVII. Положить $\gamma_0 = \gamma_0/2$ и перейти к шагу XXV.

(Если $(\nabla f_0(x^k), \bar{h}^k) = 0$ (такой случай возможен лишь на начальной стадии процесса), то следует поменять вектор g^k, изменив число ρ. Иначе на шагах XXV—XXVII будет зацикливание).

XXVIII. Вычислить число

$$\bar{\rho}_k = \min\left\{1, \delta \frac{|(g^k, h^k)|}{\|h^k\|^3}\right\}.$$

XXIX. Положить $\rho = \bar{\rho}_k$.

XXX. Вычислить $\beta_k = \text{sign}\,(h^k, g^k)$.

XXXI. Вычислить $\Delta = (f_0(x^k + \rho h^k) - f_0(x^k))/\rho^2$.

XXXII. Если $\Delta \leqslant \varepsilon \beta_k (h^k, g^k)$, то положить $\rho_k = \rho$ и перейти к шагу XXXIII; иначе положить $\rho = \rho/2$ и перейти к шагу XXXI.

XXXIII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

XXXIV. Положить $k = k + 1$ и перейти к шагу III.

Алгоритм 2А **(алгоритм вычисления вектора r^k для алгоритма 2)**

I. Выбрать произвольную последовательность $\{\lambda_i\}_{i=1}^{\infty}$, сходящуюся к нулю (рекомендуется выбрать $\lambda_i = (1/i)^{\eta}$, $\eta \in (0, 2]$).

II. Положить $\tau = 0$.

III. Положить $s = 1$.

IV. Вычислить индекс $k = \tau n + s$.

V. Вычислить вектор $r^k = \lambda_k e^s$, где e^s — s-й орт.

VI. Если $s < n$, то положить $s = s + 1$ и перейти к шагу IV; иначе положить $\tau = \tau + 1$ и перейти к шагу III.

Для алгоритма 2 справедлива теорема, аналогичная теореме 1.

Библиографические указания. Параграф написан на основании работ [320, 111, 112].

2.4. Методы сопряженных градиентов

Задача 1. Найти $\arg\min_{x\in R^n} f_0(x)$ для заданной чепрерывно дифференцируемой функции $f_0: R^n \to R^1$.

1. Общая схема алгоритмов сопряженных градиентов

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in R^n$, удовлетворяющее условиям теоремы 1; положить $k = 0$.

Основной цикл. II. Вычислить $\nabla f_0(x^k)$.

III. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Вычислить вектор движения $h^k \in Q(x^k)$, где множество $Q(x)$ задается формулой

$$Q(x) = \{h \mid (\nabla f_0(x), h) < 0, \quad h \in R^n\}.$$

V. Вычислить шаговый множитель ρ_k, удовлетворяющий условию

$$f_0(x^k + \rho_k h^k) = \min_{\rho\geqslant 0} f_0(x^k + \rho h^k).$$

VI. Вычислить следующее приближение $x^{k+1} = x^k + \rho_k h^k$.

VII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Пусть f_0 — строговыпукла и дважды непрерывно дифференцируема функция, причем для всех $x \in X_0$,*

$$X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in R^n\}$$

и всех $y \in R^n$ выполняется

$$\gamma_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x)\, y, \; y) \leqslant \gamma_2 \|y\|^2, \quad \gamma_1 > 0.$$

Тогда, если на шаге IV алгоритма 1 вектор h^k выбирать так, что для некоторого фиксированного $\alpha > 0$ выполняется неравенство

$$-(\nabla f_0(x^k), \; h^k) \geqslant \alpha \|\nabla f_0(x^k)\| \cdot \|h^k\|,$$

то алгоритм 1 порождает либо конечную последовательность $\{x^k\}_{k=0}^N$, последний элемент которой минимизирует функцию f_0 в R^n, либо бесконечную последовательность $\{x^k\}_{k=0}^\infty$, сходящуюся к точке минимума x^*.

Все варианты сопряженных градиентов отличаются только разными способами вычисления векторов h^k на шаге IV алгоритма 1 и способами вычисления шагового множителя ρ_k на шаге V алгоритма 1.

2. Метод сопряженных градиентов с восстановлением

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, натуральное число $\tau \geqslant n$ (τ — момент восстановления); положить $k = 0$.

II. Вычислить $\nabla f_0(x^0)$ и положить $g^0 = -\nabla f_0(x^0)$, $h^0 = -\nabla f_0(x^0)$.

Основной цикл. III. Вычислить шаговый множитель ρ_k, удовлетворяющий условию

$$f_0(x^k + \rho_k h^k) = \min_{\rho \geqslant 0} f_0(x^k + \rho h^k).$$

IV. Вычислить следующее приближение $x^{k+1} = x^k + \rho_k h^k$.

V. Вычислить $\nabla f_0(x^{k+1})$ и положить $g^{k+1} = -\nabla f_0(x^{k+1})$.

VI. Если $g^{k+1} = 0$, то положить $x^* = x^{k+1}$ и прекратить вычисления; иначе перейти к шагу VII.

VII. Вычислить коэффициент

$$\beta_k = w\left(\frac{k+1}{\tau}\right)\frac{(g^{k+1} - g^k,\ g^{k+1})}{(g^k,\ g^k)},$$

где $w(t) = \begin{cases} 0, & \text{если } t \text{ целое число;} \\ 1, & \text{если } t \text{ нецелое число.} \end{cases}$

VIII. Вычислить вектор

$$h^{k+1} = g^{k+1} + \beta_k h^k.$$

IX. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Если функция f_0 трижды непрерывно дифференцируема и ее матрица вторых производных $\nabla^2_{xx} f_0(x)$ удовлетворяет условию*

$$\gamma_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x)\, y,\ y) \leqslant \gamma_2 \|y\|^2, \quad 0 < \gamma_1 \leqslant \gamma_2 < \infty$$

для всех x, $y \in R^n$, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2 с произвольным начальным приближением x^0, сходится к решению x^* и существуют такое целое число $\nu > 0$ и константа $\delta_1 \in (0, \infty)$, что

$$\|x^{k+n} - x^*\| \leqslant \delta_1 \|x^k - x^*\|^2$$

для всех $k \geqslant \nu$, принадлежащих множеству $\{0, \tau, 2\tau, \ldots\}$.

Алгоритм 2'

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, натуральное число $\tau \geqslant n$ (τ — момент восстановления); положить $k = 0$.

II. Вычислить $\nabla f_0(x^0)$ и положить $g^0 = -\nabla f_0(x^0)$, $h^0 = -\nabla f_0(x^0)$.

Основной цикл. III. Вычислить шаговый множитель ρ_k, удовлетворяющий условию

$$f_0(x^k + \rho_k h^k) = \min_{\rho \geqslant 0} f_0(x^k + \rho h^k).$$

IV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

V. Вычислить $\nabla f_0(x^{k+1})$. Если $\nabla f_0(x^{k+1}) = 0$, то положить $x^* = x^{k+1}$ и прекратить вычисления; иначе перейти к шагу VI.

VI. Если k кратно τ, то перейти к шагу VII; иначе перейти к шагу VIII.

VII. Положить $g^{k+1} = -\nabla f_0(x^{k+1})$, $h^{k+1} = -\nabla f_0(x^{k+1})$ и перейти к шагу XI.

VIII. Положить $g^{k+1} = -\nabla f_0(x^{k+1})$.

IX. Вычислить коэффициент

$$\beta_k = \frac{\|g^{k+1}\|^2}{\|g^k\|^2}.$$

X. Вычислить вектор $h^{k+1} = g^{k+1} + \beta_k h^k$.

XI. Положить $k = k + 1$ и перейти к шагу III.

Для алгоритма 2' имеет место теорема, аналогичная теореме 2.

3. Реализуемые модификации алгоритмов сопряженных градиентов

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, константы $\varepsilon_0 > 0$, $\gamma_0 > 0$, $\beta \in (0, 1)$, $\beta' \in (0, 1)$, $\beta'' \in (0, 1)$ (рекомендуется $\varepsilon_0 = \cos 85°$, $\gamma_0 = \cos 5°$, $\beta = 0{,}6$, $\beta' = \beta'' = 0{,}8$), натуральное число $\tau \geqslant n$ (τ — момент восстановления); положить $k = 0$.

II. Вычислить $\nabla f_0(x^0)$.

III. Если $\nabla f_0(x^0) = 0$, то положить $x^* = x^0$ и прекратить вычисления; иначе положить $g^0 = -\nabla f_0(x^0)$, $h^0 = -\nabla f_0(x^0)$, $\varepsilon = \varepsilon_0$, $\gamma = \gamma_0$ и перейти к шагу IV.

Основной цикл. IV. Положить $x = x^k$.

V. Вычислить вектор $h = \frac{1}{\|h^k\|} h^k$.

VI. Определить функцию $\theta : \boldsymbol{R}^1 \to \boldsymbol{R}^1$

$$\theta(\mu) = f_0(x + \mu h) - f_0(x).$$

VII. Положить $\rho = 0$.

VIII. Вычислить скалярное произведение $(\nabla f_0(x + \rho h), h)$ и положить $\varphi(\rho) = (\nabla f_0(x + \rho h), h)$.

IX. Если $\varphi(\rho) = 0$, то перейти к шагу XVIII; иначе перейти к шагу X.

X. Положить $\lambda = 1$.

XI. Вычислить

$$\Delta = \theta(\rho - \lambda\varphi(\rho)) - \theta(\rho) + \frac{1}{2}\lambda\varphi^2(\rho).$$

XII. Если $\Delta \leqslant 0$, то положить $\rho = \rho - \lambda\varphi(\rho)$ и перейти к шагу XIII; иначе положить $\lambda = \beta\lambda$ и перейти к шагу XI.

XIII. Вычислить $\nabla f_0 (x + \rho h)$.

XIV. Если $\nabla f_0 (x + \rho h) = 0$, то положить $x^* = x + \rho h$ и прекратить вычисления; иначе перейти к шагу XV.

XV. Вычислить

$$\varphi(\rho) = (\nabla f_0 (x + \rho h),\ h).$$

XVI. Вычислить

$$\delta = \varphi(\rho)/\|\nabla f_0 (x + \rho h)\|.$$

XVII. Если $|\delta| \leqslant \varepsilon$, то перейти к шагу XVIII; иначе перейти к шагу X.

XVIII. Вычислить следующее приближение $x^{k+1} = x + \rho h$.

XIX. Положить $g^{k+1} = -\nabla f_0 (x + \rho h)$.

XX. Вычислить коэффициент

$$\beta_k = w\left(\frac{k+1}{\tau}\right)\frac{(g^{k+1} - g^k,\ g^k)}{\|g^k\|^2},$$

где

$$w(t) = \begin{cases} 0, & \text{если } t \text{ — целое число,} \\ 1, & \text{если } t \text{ — нецелое число.} \end{cases}$$

XXI. Вычислить вектор $h^{k+1} = g^{k+1} + \beta_k h^k$.

XXII. Положить $k = k + 1$.

XXIII. Если выполняется неравенство

$$(g^k,\ h^k) \geqslant \gamma \|g^k\| \|h^k\|,$$

то перейти к шагу IV; иначе положить $\varepsilon = \beta'\varepsilon$, $\gamma = \beta''\gamma$ и перейти к шагу IV.

Для алгоритма 3 имеет место теорема, аналогичная теореме 2.

Алгоритм 3' **(реализуемая модицикация алгоритма 2)**

Все шаги алгоритма 2, кроме шага III, остаются без изменений. Шаг III алгоритма 2 следует заменить шагом III'.

III'. Вычислить шаговый множитель ρ_k, используя для этого алгоритм 1Б или алгоритм 2Б.

Алгоритм 1Б (алгоритм вычисления шагового множителя ρ_k)

I. Выбрать константы $\rho > 0$ и $\lambda \in (0, 1/2)$.

II. Определить функции

$$\theta(\alpha,\ x^k) = f_0 (x^k + \alpha h^k) - f_0 (x^k) - \alpha (1 - \lambda) (\nabla f_0 (x^k),\ h^k);$$

$$\varphi(\alpha,\ x^k) = f_0 (x^k + \alpha h^k) - f_0 (x^k) - \alpha\lambda (\nabla f_0 (x^k),\ h^k).$$

III. Положить $\mu = \rho$.

IV. Вычислить значение $\theta(\mu,\ x^k)$.

V. Если $\theta(\mu,\ x^k) = 0$, то положить $\rho_k = \mu$ и прекратить вычисления; если $\theta(\mu,\ x^k) < 0$, то положить $\mu = \mu + \rho$ и перейти к шагу IV; если $\theta(\mu,\ x^k) > 0$, то перейти к шагу VI.

VI. Вычислить значение $\varphi(\mu,\ x^k)$.

VII. Если $\varphi(\mu, x^k) \leqslant 0$, то положить $\rho_k = \mu$ и прекратить вычисления; иначе положить $a_0 = \mu - \rho$, $b_0 = \mu$ и перейти к шагу VIII.

VIII. Положить $j = 0$.

IX. Вычислить $v_j = (a_j + b_j)/2$.

X. Вычислить значения $\theta(v_j, x^k)$, $\varphi(v_j, x^k)$.

XI. Если выполняются неравенства $\theta(v_j, x^k) \geqslant 0$ и $\varphi(v_j, x^k) \leqslant \leqslant 0$, то положить $\rho_k = v_j$ и прекратить вычисления; иначе перейти к шагу XII.

XII. Если $\theta(v_j, x^k) > 0$, то положить $a_{j+1} = a_j$, $b_{j+1} = v_j$ и перейти к шагу XIII; иначе положить $a_{j+1} = v_j$, $b_{j+1} = b_j$ и перейти к шагу XIII.

XIII. Положить $j = j + 1$ и перейти к шагу IX.

Алгоритм 2Б **(алгоритм вычисления шагового множителя ρ_k по методу золотого сечения)**

I. Выбрать произвольные константы $\varepsilon_0 > 0$, $\alpha > 0$, $\rho > 0$, $\beta \in (0, 1)$.

II. Вычислить дроби Фибоначчи $F_1 = (3 - \sqrt{5})/2$ и $F_2 = = (\sqrt{5} - 1)/2$.

III. Положить $\varepsilon = \varepsilon_0$.

IV. Определить функцию $\theta : R^1 \to R^1$

$$\theta(\mu) = f_0(x^k + \mu h^k) - f_0(x^k).$$

V. Вычислить значения $\theta(\rho)$ и $\theta(0)$.

VI. Если $\theta(\rho) \geqslant \theta(0)$, то положить $a_0 = 0$, $b_0 = \rho$ и перейти к шагу X; иначе положить $i = 0$, $\mu_0 = 0$ и перейти к шагу VII.

VII. Вычислить $\mu_{i+1} = \mu_i + \rho$.

VIII. Вычислить $\theta(\mu_{i+1})$.

IX. Если $\theta(\mu_{i+1}) \geqslant \theta(\mu_i)$, то положить $a_0 = \mu_{i-1}$, $b_0 = \mu_{i+1}$ и перейти к шагу X; иначе положить $i = i + 1$ и перейти к шагу VII.

X. Положить $j = 0$.

XI. Вычислить $l_j = b_j - a_j$.

XII. Если $l_j \leqslant \varepsilon$, то перейти к шагу XVII; иначе перейти к шагу XIII.

XIII. Вычислить

$$v_j = a_j + F_1 l_j; \quad w_j = a_j + F_2 l_j.$$

XIV. Вычислить $\theta(v_j)$, $\theta(w_j)$.

XV. Если $\theta(v_j) < \theta(w_j)$, то положить $a_{j+1} = a_j$, $b_{j+1} = w_j$ и перейти к шагу XVI; иначе положить $a_{j+1} = v_j$, $b_{j+1} = b_j$ и перейти к шагу XVI.

XVI. Положить $j = j + 1$ и перейти к шагу XI.

XVII. Положить $\bar{\mu} = (a_j + b_j)/2$.

XVIII. Если $\theta(\bar{\mu}) \leqslant -\alpha\varepsilon$, то положить $\rho_k = \bar{\mu}$ и прекратить вычисления; иначе положить $\varepsilon = \beta\varepsilon$ и перейти к шагу V.

Алгоритм 3'' (реализуемая модификация алгоритма 2')

Все шаги алгоритма 2′, кроме шага III, остаются без изменений. Шаг III алгоритма 2′ следует заменить на шаг III″.

III″. Вычислить шаговый множитель ρ_k, используя для этого алгоритм 1Б или алгоритм 2Б.

4. Метод сопряженных градиентов для минимизации квадратичных функций

Задача 4. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной квадратичной функции $f_0(x) \triangleq {}^1/_2\,(Ax, x) + (b, x) + \alpha$.

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$ и положить $k = 0$.

II. Вычислить $Ax^0 + b$ и положить $h^0 = -(Ax^0 + b)$, $z^0 = -h^0$.

III. Если $h^0 = 0$, то положить $x^* = x^0$ и прекратить вычисления; иначе перейти к шагу IV.

Основной цикл. IV. Если $k = 0$, то перейти к шагу VII; иначе вычислить вектор g^{k-1}, $g^{k-1} = A(x^k - x^{k-1}) = \rho_{k-1}Ah^{k-1}$ и перейти к шагу V.

V. Вычислить коэффициент β_k по одной из эквивалентных формул

$$\beta_k = -\frac{(z^k, g^{k-1})}{(h^{k-1}, z^{k-1})}; \quad \beta_k = -\frac{(z^k, z^k)}{(h^{k-1}, z^{k-1})}; \quad \beta_k = \frac{(z^k, z^k)}{(z^{k-1}, z^{k-1})}.$$

VI. Вычислить вектор $h^k = -z^k + \beta_k h^{k-1}$.

VII. Вычислить шаговый множитель ρ_k по одной из эквивалентных формул

$$\rho_k = -\frac{(z^0, h^k)}{(Ah^k, h^k)}; \quad \rho_k = -\frac{(z^k, h^k)}{(Ah^k, h^k)}.$$

VIII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

IX. Вычислить $Ax^{k+1} + b$ и положить $z^{k+1} = Ax^{k+1} + b$.

X. Если $z^{k+1} = 0$, то положить $x^* = x^{k+1}$ и прекратить вычисления; иначе перейти к шагу XI.

XI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 4. *Если A — строго положительно-определенная симметричная матрица, то алгоритм 4 решает задачу 4 за число итераций, не превосходящее n и (i) — для всех i, $0 \leqslant i \leqslant n-1$, выполняется неравенство $(Ax^i + b, h^i) \neq 0$, если $Ax^i + b \neq 0$; (ii) — точка x^i ($i = 1, \ldots, n-1$) является точкой минимума квадратичной функции f_0 на подпространстве, образованном векторами $h^0, \ldots, h^{i-1}$ и проходящем через точку x^0.*

5. Реализуемая модификация алгоритма с переменной метрикой

Алгоритм 5

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, целое число α $(1 \leqslant \alpha \leqslant 10)$, константу $\beta \in (1/2, 4/5)$ и положить $k = 0$, $j = 0$.

II. Вычислить $\nabla f_0(x^0)$, если $\nabla f_0(x^0) = 0$, то положить $x^* = x^0$ и прекратить вычисления; иначе перейти к шагу III.

III. Положить $H_0 = I$ (I — единичная $n \times n$-матрица), $g^0 = \nabla f_0(x^0)$.

Основной цикл. IV. Вычислить вектор $h^k = H_k g^k$.

V. Если число k кратно α, то положить $j = j + 1$, $\alpha = 3\alpha$ и перейти к шагу VI; иначе перейти к шагу VI.

VI. Положить $x = x^k$, $h = h^k$, $i = 0$.

VII. Определить функцию $\theta : R^1 \to R^1$

$$\theta(\alpha) = f_0(x + \alpha h) - f_0(x).$$

VIII. Положить $\rho = 0$.

IX. Вычислить $\varphi(\rho) = (\nabla f_0(x + \rho h), h)$.

X. Если $\varphi(\rho) = 0$, то положить $\rho_k = \rho$ и перейти к шагу XV; иначе перейти к шагу XI.

XI. Положить $\lambda = 1$.

XII. Вычислить $\Delta = \theta(\rho - \lambda\varphi(\rho)) - \theta(\rho) + \frac{1}{2}\lambda\varphi^2(\rho)$.

XIII. Если $\Delta \leqslant 0$, то положить $i = i + 1$ и перейти к шагу XIV; иначе положить $\lambda = \beta\lambda$ и перейти к шагу XII.

XIV. Если $i < j$, то положить $\rho = \rho - \lambda\varphi(\rho)$ и перейти к шагу IX; иначе положить $\rho_k = \rho - \lambda\varphi(\rho)$ и перейти к шагу XV.

XV. Вычислить следующее приближение $x^{k+1} = x + \rho_k h$.

XVI. Вычислить $\nabla f_0(x^{k+1})$ и положить $g^{k+1} = \nabla f_0(x^{k+1})$.

XVII. Если $g^{k+1} = 0$, то положить $x^* = x^{k+1}$ и прекратить вычисления; иначе перейти к шагу XVIII.

XVIII. Вычислить векторы

$$r^k = g^{k+1} - g^k; \quad z^k = x^{k+1} - x^k.$$

XIX. Вычислить матрицу

$$H_{k+1} = H_k - \frac{H_k r^k (H_k r^k)^T}{(H_k r^k, r^k)} + \frac{z^k (z^k)^T}{(r^k, z^k)}.$$

XX. Положить $k = k + 1$ и перейти к шагу IV.

Библиографические указания. Параграф написан на основании работ [285], [320]. Методы сопряженных градиентов и алгоритмы с переменной метрикой решения задач безусловной оптимизации исследовались также в работах [229, 494, 230, 291, 292, 539, 471, 506, 492, 470, 390, 454, 362].

В работе [542] доказываются теоремы о конечной сходимости варианта метода сопряженных градиентов для минимизации начального направления поиска. В работах [482, 484, 534, 466] изучаются методы типа переменной метрики.

2.5. Методы сопряженных направлений

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной непрерывно дифференцируемой функции $f_0 : R^n \to R^1$.

Предположение 1. Функция f_0 дважды непрерывно дифференцируема и ее матрица вторых производных $\nabla^2_{xx} f_0(x)$ удовлетворяет условию

$$\gamma_1 \| y \|^2 \leqslant (\nabla^2_{xx} f_0(x) y, y) \leqslant \gamma_2 \| y \|^2, \quad \gamma_2 \geqslant \gamma_1 > 0,$$

при всех $x, y \in R^n$.

Определение 1. Векторы $p^0, p^1, \ldots, p^{n-1}$ называются сопряженными, или A-ортогональными, если $(p^i, Ap^j) = 0$ при $i \neq j$, где A — строго положительно-определенная матрица.

1. Метод сопряженных направлений с восстановлением матрицы

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, произвольную симметричную строго положительно-определенную матрицу H_0, удовлетворяющую условию $\gamma_3 \| y \|^2 \leqslant (H_0 y, y) \leqslant \gamma_4 \| y \|^2$, $\gamma_4 \geqslant \gamma_3 > 0$, $\forall y \in R^n$, (в частности, можно выбрать $H_0 = \mathrm{I}$, где I — единичная $n \times n$-матрица); положить $k = 0$.

Основной цикл. II. Вычислить $\nabla f_0(x^k)$.

III. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Если $k = 0$, то перейти к шагу VIII; иначе перейти к шагу V.

V. Если k делится нацело на n, то положить $H_k = H_0$ и перейти к шагу VIII; иначе перейти к шагу VI.

VI. Вычислить векторы

$$r^{k-1} = x^k - x^{k-1} = \rho_{k-1} h^{k-1};$$

$$g^{k-1} = \nabla f_0(x^k) - \nabla f_0(x^{k-1}).$$

VII. Вычислить матрицу H_k по одной из приведенных ниже формул:

$$H_k = H_{k-1} + \frac{r^{k-1} (r^{k-1})^T}{(r^{k-1}, g^{k-1})} - \frac{H_{k-1} g^{k-1} (g^{k-1})^T H_{k-1}}{(H^T_{k-1} g^{k-1}, g^{k-1})}; \tag{2.3}$$

$$H_k = H_{k-1} + (r^{k-1} - H_{k-1} g^{k-1}) \frac{(r^{k-1})^T}{(r^{k-1}, g^{k-1})}; \tag{2.4}$$

$$H_k = H_{k-1} + (r^{k-1} - H_0 g^{k-1}) \frac{(r^{k-1})^T}{(r^{k-1}, g^{k-1})}; \tag{2.5}$$

$$H_k = H_{k-1} - \frac{H_{k-1} g^{k-1} (r^{k-1})^T}{(r^{k-1}, g^{k-1})}; \quad (2.6)$$

$$H_k = H_{k-1} - \frac{H_0 g^{k-1} (r^{k-1})^T}{(r^{k-1}, g^{k-1})}; \quad (2.7)$$

$$H_k = H_0 - \frac{H_0 \nabla f_0 (x^k) (H_0 \nabla f_0 (x^k) + h^{k-1})^T}{(H_0 \nabla f_0 (x^k), \nabla f_0 (x^k)) - (h^{k-1}, \nabla f_0 (x^{k-1}))}; \quad (2.8)$$

$$H_k = H_0 + \frac{H_0 g^{k-1} (h^{k-1})^T}{(h^{k-1}, \nabla f_0 (x^{k-1}))}; \quad (2.9)$$

$$H_k = H_0 + \frac{H_0 \nabla f_0 (x^k) (h^{k-1})^T}{(h^{k-1}, \nabla f_0 (x^{k-1}))}; \quad (2.10)$$

VIII. Вычислить вектор движения h^k к следующему приближению x^{k+1}

$$h^k = -H_k^T \nabla f_0 (x^k).$$

IX. Вычислить шаговый множитель ρ_k из условия

$$f_0 (x^k + \rho_k h^k) = \min_{\rho \geqslant 0} f_0 (x^k + \rho h^k)$$

X. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho^k h^k.$$

XI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Если выполнено предположение* 1, *то бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом* 1 *(где матрица* H_k *вычисляется по одной из формул* (2.3) — (2.10)), *сходится к решению* x^* *со сверхлинейной скоростью*

$$\| x^{(i+1)n} - x^* \| \leqslant \lambda_{in} \| x^{in} - x^* \|, \quad i = 0, 1, \ldots,$$

где $\lambda_{in} \to 0$ *при* $i \to \infty$.

Замечание 1. Если на какой-либо итерации начальной стадии процесса минимизации по алгоритму 1 вектор $h^k = 0$, то необходимо начать процесс заново, восстановив матрицу H_0.

2. Метод сопряженных направлений без восстановления матрицы

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, произвольную симметричную строго положительно-определенную матрицу H_0, удовлетворяющую условию

$$\gamma_3 \| y \|^2 \leqslant (H_0 y, y) \leqslant \gamma_4 \| y \|^2, \quad \gamma_4 \geqslant \gamma_3 > 0, \quad \forall y \in R^n$$

(в частности, можно выбрать $H_0 = I$, где I — единичная $n \times n$-матрица); положить $k = 0$.

Основной цикл. II. Вычислить $\nabla f_0 (x^k)$, если $\nabla f_0 (x^k) =$

$= 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу III.

III. Если $k = 0$, то перейти к шагу VI; иначе перейти к шагу IV.

IV. Вычислить векторы

$$r^{k-1} = x^k - x^{k-1} = \rho_{k-1} h^{k-1};$$

$$g^{k-1} = \nabla f_0(x^k) - \nabla f_0(x^{k-1}).$$

V. Вычислить матрицу H_k по одной из приведенных ниже формул:

$$H_k = H_{k-1} + \frac{r^{k-1}(r^{k-1})^T}{(r^{k-1}, g^{k-1})} - \frac{H_{k-1} g^{k-1} (g^{k-1})^T H_{k-1}}{(H_{k-1}^T g^{k-1}, g^{k-1})}; \quad (2.11)$$

$$H_k = H_{k-1} + (r^{k-1} - H_{k-1} g^{k-1}) \frac{(g^{k-1})^T}{(r^{k-1}, g^{k-1})}; \quad (2.12)$$

$$H_k = H_{k-1} + (r^{k-1} - H_0 g^{k-1}) \frac{(g^{k-1})^T}{(r^{k-1}, g^{k-1})}; \quad (2.13)$$

$$H_k = H_0 + \frac{H_0 g^{k-1} (h^{k-1})^T}{(h^{k-1}, \nabla f_0(x^{k-1}))}. \quad (2.14)$$

VI. Вычислить вектор движения h^k к следующему приближению x^{k+1}

$$h^k = -H_k^T \nabla f_0(x^k).$$

VII. Вычислить шаговый множитель ρ_k из условия

$$f_0(x^k + \rho_k h^k) = \min_{\rho \geq 0} f_0(x^k + \rho h^k).$$

VIII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

IX. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Пусть выполняются предположения 1. Тогда:*

1) *если первые m точек ($m < \infty$ и зависит от точки x^0) бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$ вычислены с помощью метода сопряженных направлений с восстановлением матрицы H_k (т. е. с помощью алгоритма* 1), *а остальные точки этой последовательности — с помощью метода сопряженных направлений без восстановления (т. е. с помощью алгоритма* 2), *то последовательность $\{x^k\}_{k=0}^{\infty}$ сходится к решению x^* при произвольном выборе начального приближения x^0;*

2) *если начальное приближение x^0 выбрано в достаточно малой окрестности точки минимума x^*, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, сходится к решению x^*;*

3) *бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, где матрица H_k вычисляется по формуле* (2.14), *неза-*

висимо от выбора начального приближения x^0 сходится к решению x^* со сверхлинейной скоростью

$$\|x^{(i+1)n} - x^*\| \leqslant \lambda_{in} \|x^{in} - x^*\|, \quad i = 0, 1, \ldots,$$

где $\lambda_{in} \to 0$ при $i \to \infty$.

Замечание 2. Из оценок скорости сходимости методов сопряженных направлений следует, что n итераций метода сопряженных направлений эквивалентны (в смысле скорости сходимости) одной итерации методов типа Ньютона — Канторовича или методов двойственных направлений. Однако они значительно превосходят по скорости сходимости градиентные методы.

С теоретической точки зрения методы сопряженных направлений без восстановления матрицы предпочтительнее в том случае, когда $H_{in} \to (\nabla^2_{xx} f_0 (x^{in}))^{-1}$ при $i \to \infty$. К настоящему времени установить строгое выполнение этого условия не удалось. Однако можно предполагать, что такое условие будет выполняться для методов, которые при минимизации квадратичных функций $\frac{1}{2}(Ax, x) + (b, x) + \gamma$ дают $H_n = A^{-1}$.

В [188] рекомендуется применять алгоритм 1 с восстановлением матрицы, если матрица H_k вычисляется по формулам (2.6) — (2.10), а алгоритм 2 без восстановления матрицы, когда матрица H_k вычисляется по формулам (2.3) — (2.5).

3. Минимизация квадратичных функций с помощью метода сопряженных направлений

Задача 3. Найти $\arg\min_{x \in R^n} f_0$ для заданной квадратичной функции $f_0(x) \triangleq \frac{1}{2}(Ax, x) + (b, x) + \alpha$.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, произвольную строго положительно-определенную матрицу H_0 (в частности, можно выбрать $H_0 = I$, где I — единичная $n \times n$-матрица); положить $k = 0$.

Основной цикл. II. Вычислить $Ax^k + b$ и положить

$$z^k = Ax^k + b.$$

Если $z^k = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу III;

III. Если $k = 0$, то перейти к шагу VI; иначе перейти к шагу IV.

IV. Вычислить векторы

$$r^{k-1} = x^k - x^{k-1} = \rho_{k-1} h^{k-1}; \quad g^{k-1} = Ar^{k-1}.$$

V. Вычислить матрицу H_k по любой из формул (2.3) — (2.10).

VI. Вычислить вектор движения h^k к следующему приближению x^{k+1}

$$h^k = -H_k^T z^k.$$

VII. Вычислить шаговый множитель ρ_k по одной из формул

$$\rho_k = -\frac{(z^0, h^k)}{(Ah^k, h^k)}; \quad \rho_k = -\frac{(z^k, h^k)}{(Ah^k, h^k)}.$$

VIII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

IX. Положить $k = k + 1$ и перейти к шагу II.

Теорема 3. *Если A — строго положительно-определенная симметричная матрица с постоянными членами (т. е. $(Ax, x) > 0$ при любом $x \neq 0$), то алгоритм 3 решает задачу 3 за число итераций, не превосходящее n, и (i) — точки $x^0, x^1, \ldots, x^n$, порожденные алгоритмом 3 при различных способах вычисления матриц H_k, одни и те же; (ii) — для всех i, $0 \leqslant i \leqslant n-1$ выполняется условие $(Ax^i + b, h^i) \neq 0$, если $Ax^i + b \neq 0$; (iii) — точка x^i, $i = 1, \ldots, n-1$, является точкой минимума квадратичной функции f_0 на подпространстве, образованном векторами $h^0, h^1, \ldots, h^{i-1}$ и проходящем через точку x^0; (iv) — если в алгоритме 3 матрицу H_k вычислять: по формулам (2.3) — (2.5), то $H_n = A^{-1}$; по формулам (2.8), (2.10), то $H_n = H_0$; по формуле (2.6), то $H_n = 0$; по формуле (2.9), то $H_n \neq H_0$.*

Ниже приводится метод сопряженных направлений, в котором непосредственно находят вектор движения h^k без вычисления матрицы H_k.

Алгоритм 3'

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, произвольную строго положительно-определенную симметричную матрицу H_0 (в частности, можно выбрать $H_0 = I$, где I — единичная $n \times n$-матрица); положить $k = 0$.

Основной цикл. II. Вычислить $Ax^k + b$ и положить $z^k = Ax^k + b$.

III. Если $z^k = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Если $k = 0$, то вычислить вектор h^0 по формуле $h^0 = -H_0 z^0$ и перейти к шагу V; иначе вычислить вектор движения h^k по любому из алгоритмов 1А, 2А, 3А, 4А и перейти к шагу V.

V. Вычислить шаговый множитель ρ_k по одной из эквивалентных формул

$$\rho_k = -\frac{(z^0, h^k)}{(Ah^k, h^k)}; \quad \rho_k = -\frac{(z^k, h^k)}{(Ah^k, h^k)}.$$

VI. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

VII. Положить $k = k + 1$ и перейти к шагу II.

Алгоритм 1A (алгоритм вычисления вектора h^k)

I. Вычислить вектор $g^{k-1} = z^k - z^{k-1}$.

II. Вычислить коэффициент β_k по одной из эквивалентных формул

$$\beta_k = 1 - \frac{(H_0 z^k,\ g^{k-1})}{(H_0 z^k,\ z^k) - (h^{k-1},\ z^{k-1})};$$

$$\beta_k = - \frac{(h^{k-1},\ z^{k-1})}{(H_0 z^k,\ z^k) - (h^{k-1},\ z^{k-1})}.$$

III. Вычислить вектор

$$h^k = -\beta_k H_0 z^k + (1 - \beta_k) h^{k-1}.$$

Алгоритм 2A **(алгоритм вычисления вектора h^k)**

I. Вычислить коэффициент

$$\beta_k = \frac{(H_0 z^k,\ z^k)}{(H_0 z^k,\ z^k) - (h^{k-1},\ z^{k-1})}.$$

II. Вычислить вектор

$$h^k = -H_0 z^k + \beta_k (H_0 z^k + h^{k-1}).$$

Алгоритм 3A (алгоритм вычисления вектора h^k)

I. Вычислить вектор $g^{k-1} = z^k - z^{k-1}$.

II. Вычислить коэффициент β_k по одной из формул

$$\beta_k = \frac{(H_0 z^k,\ g^{k-1})}{(h^{k-1},\ g^{k-1})}; \quad \beta_k = - \frac{(H_0 z^k,\ g^{k-1})}{(h^{k-1},\ z^{k-1})}.$$

III. Вычислить вектор

$$h^k = -H_0 z^k + \beta_k h^{k-1}.$$

Алгоритм 4A (алгоритм вычисления вектора h^k)

I. Вычислить коэффициент β_k по одной из эквивалентных формул

$$\beta_k = - \frac{(H_0 z^k,\ z^k)}{(h^{k-1},\ z^{k-1})}; \quad \beta_k = \frac{(H_0 z^k,\ z^k)}{(H_0 z^{k-1},\ z^{k-1})}.$$

II. Вычислить вектор

$$h^k = -H_0 z^k + \beta_k h^{k-1}.$$

Теорема 3'. *Если A — строго положительно-определенная симметричная матрица с постоянными членами, то алгоритм 3' решает задачу 3 за число итераций, не превосходящее n, и для последовательности $x^0, \ldots, x^n$, порожденной алгоритмом 3', справедливы утверждения (i), (ii), (iii) теоремы 3.*

Методы сопряженных направлений могут быть применены и для минимизации выпуклой квадратичной функции

$$f_0(x) \triangleq {}^1/_2 (Ax,\ x) + (b,\ x) + \alpha,$$

где $(Ax, x) \geqslant 0$ при $x \neq 0$. При этом следует учитывать два случая.

1. Минимум квадратичной функции f_0 существует. Тогда алгоритм 3 и алгоритм 3′ решают задачу минимизации f_0 за число итераций, не превышающее n.

2. Функция f_0 не достигает минимума. Тогда при некотором k, $0 \leqslant k \leqslant n-1$ будет выполняться условие $(Ah^k, h^k) = 0$, что влечет $\rho_k = \infty$.

4. Модифицированный метод сопряженных направлений, не требующий вычисления производных

Алгоритм 4

Начало. I. Выбрать: начальное приближение $x^{1,0} \in R^n$, удовлетворяющее условиям теоремы 4; ортонормированный координатный базис $h^{1,1}, h^{1,2}, \ldots, h^{1,n}$; произвольную константу $\varepsilon \in (0, 1)$ (рекомендуется выбрать $\varepsilon \in [10^{-10}, 10^{-3}]$).

II. Положить $k = 1$.

Основной цикл. III. Положить $i = 1$.

IV. Вычислить шаговый множитель $\rho_{k,i}$ из условия

$$f_0(x^{k,i-1} + \rho_{k,i}h^{k,i}) = \min_{\rho} f_0(x^{k,i-1} + \rho h^{k,i}).$$

V. Вычислить следующее приближение

$$x^{k,i} = x^{k,i-1} + \rho_{k,i}h^{k,i}.$$

VI. Если $i < n$, то положить $i = i + 1$ и перейти к шагу IV; иначе перейти к шагу VII.

VII. Вычислить $\gamma_k = \| x^{k,n} - x^{k,0} \|$.

VIII. Вычислить вектор

$$h^{k,n+1} = \frac{1}{\gamma_k}(x^{k,n} - x^{k,0}).$$

IX. Вычислить шаговый множитель $\rho_{k,n+1}$ из условия

$$f_0(x^{k,h} + \rho_{k,n+1}h^{k,n+1}) = \min_{\rho} f_0(x^{k,n} + \rho h^{k,n+1}).$$

X. Вычислить следующее приближение

$$x^{k,n+1} = x^{k,n} + \rho_{k,n+1}h^{k,n+1}.$$

XI. Найти индекс $s \in [1 : n]$, удовлетворяющий условию

$$\rho_{k,s} = \max\{\rho_{k,1}, \rho_{k,2}, \ldots, \rho_{k,n}\}.$$

XII. Если $k = 1$, то вычислить определитель Δ_k, столбцами которого являются векторы $h^{k,1}, h^{k,2}, \ldots, h^{k,n}$; иначе перейти к шагу XIII.

XIII. Если $(\rho_{k,s}\Delta_k/\gamma_k) \geqslant \varepsilon$, то перейти к шагу XIV; иначе перейти к шагу XVIII.

XIV. Положить $j = 1$.

XV. Если $j = s$, то положить $h^{k+1,j} = h^{k,n+1}$ и перейти к шагу XVI; иначе положить $h^{k+1,j} = h^{k,j}$ и перейти к шагу XVI.

XVI. Если $j < n$, то положить $j = j + 1$ и перейти к шагу XV; иначе перейти к шагу XVII.

XVII. Вычислить $\Delta_{k+1} = \rho_{k,s}\Delta_k/\gamma_k$ и перейти к шагу XXII.

XVIII. Положить $j = 1$.

XIX. Положить $h^{k+1,j} = h^{k,j}$.

XX. Если $j < n$, то положить $j = j + 1$ и перейти к шагу XIX; иначе перейти к шагу XXI.

XXI. Положить $\Delta_{k+1} = \Delta_k$ и перейти к шагу XXII.

XXII. Положить $x^{k+1,0} = x^{k,n+1}$.

XXIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 4. *Если выполнены условия: (i) — функция f_0 — непрерывно дифференцируема; (ii) — функция f_0 — строговыпуклая; (iii) — начальное приближение $x^{1,0}$ в алгоритме 4 таково, что множество*

$$X_0 = \{x \mid f_0(x) \leqslant f_0(x^{1,0}), \quad x \in R^n\}$$

ограничено, то последовательность $x^{k,i}$, $i = 0, 1, \ldots, n$; $k = 1, 2, \ldots$ ..., порожденная алгоритмом 4, сходится к единственной точке минимума x^* функции f_0.

Библиографические указания. Параграф основан на работах [320, 108, 110, 112, 498]. Методы сопряженных направлений исследовались также в работах [109, 295, 540, 570, 470, 362].

2.6. Методы псевдообратных операторов

Задача 0. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположение 0. (*i*) — функция $f_0(x)$ дважды дифференцируемая по Фреше; (*ii*) — градиент $\nabla f_0(x)$ функции $f_0(x)$ удовлетворяет в R^n условию Липшица с константой $\gamma_1 \geqslant 0$; (*iii*) — матрица вторых производных $\Delta^2_{xx} f_0(x)$ (гессиан) функции $f_0(x)$ удовлетворяет в R^n условию Липшица с константой $\gamma_2 \geqslant 0$.

Методы псевдообратных операторов по существу являются квазиньютоновскими. Скорость сходимости этих методов — сверхлинейная. В k-й итерации за направление движения к следующему приближению x^{k+1} выбирается вектор $(-B_k \nabla f_0(x^k))$, где B_k — матрица, аппроксимирующая обратный гессиан функции $f_0(x)$ в точке x^k. Вычисление матрицы B_k основано на алгоритмах устойчивого псевдообращения прямоугольных матриц. Достоинством методов псевдообратных операторов является то, что они не требуют вычисления матрицы вторых производных функции $f_0(x)$, устойчивы к возмущениям, которые возникают из-за ошибок аппроксимации и дискретизации вычислений, обладают сверхлинейной скоростью сходимости и тогда, когда гессиан вырождается.

1. Основной алгоритм

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать значение параметра m, $m \geqslant n$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить $\nabla f_0(x^k)$.

V. Если $k \geqslant m$, то перейти к шагу VI; иначе положить $x^{k+1} = x^k - \nabla f_0(x^k)$ и перейти к шагу XI.

VI. Сформировать вспомогательную $n \times m$-матрицу $A_{k,m}$, j-м столбцом которой является вектор

$$a^j = x^{k-(m-j)} - x^{k-(m-j+1)}, \quad j = 1, \ldots, m.$$

VII. Сформировать вспомогательную $n \times m$-матрицу $H_{k,m}$, j-м столбцом которой является вектор

$$h^j = \nabla f_0(x^{k-(m-j)}) - \nabla f_0(x^{k-(m-j+1)}), \quad j = 1, \ldots, m.$$

VIII. Используя алгоритм 1′ устойчивого псевдообращения конечномерных прямоугольных матриц, вычислить псевдообратные матрицы $A_{k,m}^+$ и $(H_{k,m}A_{k,m}^+)^+$, где символ C^+ обозначает матрицу, псевдообратную к матрице C.

IX. Вычислить матрицу B_k, аппроксимирующую обратный гессиан

$$B_k = (H_{k,m}A_{k,m}^+)^+ + I - H_{k,m}A_{k,m}^+(H_{k,m}A_{k,m}^+)^+.$$

X. Вычислить следующее приближение

$$x^{k+1} = x^k - B_k\nabla f_0(x^k).$$

XI. Положить $k = k + 1$ и перейти к шагу IV.

Ниже приводится алгоритм $\Pi_\varepsilon^{(0)}$ для псевдообращения $n \times m$-матрицы C_n, состоящей из строк a^i, $i = 1, \ldots, n$,

$$(a^i = (a_1^i, a_2^i, \ldots, a_m^i)).$$

***Алгоритм 1′* (регуляризованный рекуррентный алгоритм $\Pi_\varepsilon^{(0)}$ устойчивого псевдообращения конечномерных прямоугольных матриц)**

Начало. I. Выбрать параметр регуляризации $\varepsilon \geqslant 0$.

II. Вычислить $\|a^1\|_1 = \sum_{j=1}^{m} |a_j^1|$.

III. Если $\|a^1\|_1 > \varepsilon$, то вычислить матрицу размера $n \times 1$

$$C_1^+ = (a^1)^T (a^1 (a^1)^T)^{-1}$$

и перейти к шагу IV; если $\|a^1\|_1 \leqslant \varepsilon$, то положить $C_1^+ = 0$ и перейти к шагу IV.

IV. Положить $i = 1$.

Основной цикл. V. Если $i < n$, то перейти к шагу VI; иначе прекратить вычисления.

VI. Вычислить вектор-строку d^i размера i

$$d^i = a^{i+1} C_i^{\#}.$$

VII. Вычислить вектор-строку b^i размера m

$$b^i = a^{i+1} - d^i C_i,$$

где C_i — матрица размера $i \times m$, состоящая из первых i строк исходной матрицы C_n.

VIII. Вычислить $\| b^i \|_1 = \sum_{j=1}^{m} | b_j^i |$, если $\| b^i \|_1 > \varepsilon$, то перейти к шагу IX; иначе перейти к шагу XI.

IX. Вычислить вектор-столбец g^i размера m

$$g^i = (I - C_i^{\#} C_i)(a^{i+1})^T.$$

X. Вычислить вектор-столбец r^{i+1} размера m

$$r^{i+1} = g^i (b^i g^i)^{-1}$$

и перейти к шагу XII.

XI. Вычислить вектор-столбец r^{i+1} размера m

$$r^{i+1} = C_i^{\#} (d^i)^T (1 + d^i (d^i)^T)^{-1}$$

и перейти к шагу XII.

XII. Вычислить матрицу S_i размера $m \times i$

$$S_i = C_i^{\#} - r^{i+1} d^i.$$

XIII. Составить матрицу $C_{i+1}^{\#}$ размера $m \times (i + 1)$, состоящую из матрицы S_i и вектор-столбца r^{i+1}

$$C_{i+1}^{\#} = (S_i \vdots r^{i+1}).$$

XIV. Положить $i = i + 1$ и перейти к шагу V.

При $\varepsilon = 0$ матрица $C_n^{\#}$, порождаемая алгоритмом 1′, всегда совпадает с псевдообратной к C_n, т. е. $C_n^{\#} = C_n^{+}$, а алгоритм 1′ есть метод Гревилля псевдообращения прямоугольных матриц.

При подходящем (малом) $\varepsilon > 0$ алгоритм $\Pi_\varepsilon^{(0)}$ устойчив к малым возмущениям вычислений, а также к возмущениям элементов псевдобращаемой матрицы C_n (метод Гревилля очень чувствителен к ошибкам вычислений). Эта устойчивость имеет место для широкого класса матриц даже неполного ранга.

Определение 1. Последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится сверхлинейно к стационарной точке x^* функции $f_0(x)$, если при некотором $k_1 < \infty$ достигается $Px^{k_1} = Px^*$ или при всех $k \geqslant 0$ — $Px^k \neq Px^*$ и $\lim_{k \to \infty} \| P(x^{k+1} - x^*) \| / \| P(x^k - x^*) \| = 0$, где P — ортопроектор на образ $R(\nabla_{xx}^2 f_0(x^*))$ гессиана $\nabla_{xx}^2 f_0(x^*)$ в точке x^*.

Теорема 1. Пусть выполняются предположения 0 *и следующие условия:* (*iv*) — *последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом* 1, *сходится к стационарной точке* x^* *функции* $f_0(x)$; (*v*) — *при каждом* $k \geqslant m$ *матрица* $A_{k,m}$ *имеет образ* $R(A_{k,m})$, *который содержит образ* $R(\nabla^2_{xx} f_0(x^k))$ *матрицы вторых производных, найденной в точке* x^k; (*vi*) — *при каждом* $k \geqslant 0$ *справедливо*

$$Px^k \neq Px^*,$$

где P — ортопроектор на образ $R(\nabla^2_{xx} f_0(x^*))$ гессиана $\nabla^2_{xx} f_0(x^*)$ в точке x^*;

(*vii*) — *при каждом* $k \geqslant 0$ *справедливо*

$$R(B_k^T) \supset R(P).$$

Тогда достаточными условиями сверхлинейной скорости сходимости являются

$$\lim_{k\to\infty} \| P(B_k^+ - \nabla^2_{xx} f_0(x^*))(x^{k+1} - x^k) \| / \| P(x^{k+1} - x^k) \| = 0 \qquad (2.15)$$

и выполнение условий в некоторой окрестности точки x^*

$$\| \nabla^2_{xx} f_0(x^k + \tau(x^{k+1} - x^k)) P - P \nabla^2_{xx} f_0(x^k + \tau(x^{k+1} - x^k)) \| \leqslant$$
$$\leqslant \beta_1 \| P(x^{k+1} - x^k) \|, \quad 0 < \beta_1 < \infty, \quad \forall \tau \in [0, 1]. \qquad (2.16)$$

Если для каждого x^{k+1} *имеет место неравенство* (2.16) *и*

$$\| \nabla^2_{xx} f_0(x^* + \tau(x^{k+1} - x^*)) P - P \nabla^2_{xx} f_0(x^* + \tau(x^{k+1} - x^*)) \| \leqslant$$
$$\leqslant \beta_1 \| P(x^{k+1} - x^*) \|, \quad 0 < \beta_1 < \infty, \quad \forall \tau \in [0, 1],$$

то (2.15) *является и необходимым условием.*

Теорема 1'. Пусть выполняются предположения 0, условия (*iv*), (*vi*) *теоремы 1, а в некоторой окрестности точки* x^* *выполняется неравенство* (2.16), $R(A_{k,m}) \supset R(\nabla^2_{xx} f_0(x^k))$ *и пусть матрица* B_k *на шаге* IX *алгоритма 1 вычисляется по формуле*

$$B_k = (H_{k,m} A_{k,m}^+)^{\#} + I - H_{k,m} A_{k,m}^+ (H_{k,m} A_{k,m}^+)^{\#},$$

где псевдообращение $G^{\#}$ *означает, что оно ведется алгоритмом* 1' *устойчивого псевдообращения* $\Pi^{(0)}_{\varepsilon_k}$ *с* ε_k, *удовлетворяющим условиям*

$$\max_{j=k-m+1,\ldots,k} \{ \| x^k - x^j \|, \| x^j - x^{j-1} \| \} \leqslant \beta_2 \varepsilon_k^{2+\beta};$$

$$\lim_{k\to\infty} \varepsilon_k = 0,$$

(здесь β_2, β — *некоторые положительные постоянные), тогда* $\{x^k\}_{k=0}^{\infty}$ *сходится сверхлинейно.*

Определение 2. Для прямоугольной матрицы C псевдособственными числами называют числа λ_i такие, что λ_i^2 являются собственными числами матрицы $C^T C$. Отношение $\nu = \lambda_{max}/\lambda_{min}$, где λ_{max}, λ_{min} — максимальное и минимальное из ненулевых псевдособственных чисел матрицы C, называют мерой обусловленности матрицы C. Если ν — велико, то говорят, что матрица C плохо обусловлена.

Алгоритм 1′ неприменим к псевдообращению плохо обусловленных матриц. Ниже приводится модификация алгоритма $\Pi_\varepsilon^{(0)}$, основанная на последовательном уточнении псевдообращаемой матрицы, которая позволяет псевдообращать и плохо обусловленные, даже вырожденные матрицы. Алгоритм 2 служит для псевдообращения матрицы C_n размера $n \times m$, состоящей из строк a^i, $i = 1, \ldots, n$. На выходе алгоритма 2 получается матрица $C_n^{\#}$, которая принимается за псевдообратную к C_n.

***Алгоритм 2* (алгоритм $\Pi^{(0)}$ (ε, δ, μ, ρ) устойчивого псевдообращения плохо обусловленных матриц)**

Начало. I. Выбрать произвольные константы (параметры алгоритма) $\varepsilon > 0$, $\delta > 0$, $\mu > 0$, $\rho > 0$ (рекомендуется выбрать $\mu \ll \rho$, δ — «малое» число).

II. Вычислить $\| a^1 \|_1$.

(Норма $\| b \|_1$ вектора $b = (b_1, \ldots, b_m)$ вычисляется по формуле

$$\| b \|_1 = \sum_{j=1}^{m} | b_j |).$$

III. Если $\| a^1 \|_1 > \varepsilon$, то вычислить матрицу $C_1^{\#}$ размера $m \times 1$

$$C_1^{\#} = (a^1)^T (a^1 (a^1)^T)^{-1}$$

и перейти к шагу IV; если $\| a^1 \|_1 \leqslant \varepsilon$, то положить $C_1^{\#} = 0$, где 0 — ноль-матрица размера $m \times 1$, и перейти к шагу IV.

IV. Вычислить матрицу G_1 размера $m \times m$

$$G_1 = I - C_1^{\#} C_1.$$

V. Вычислить матрицу Q_1 размера $m \times m$

$$Q_1 = C_1^{\#} (C_1^{\#})^T.$$

VI. Положить $i = 1$.

Основной цикл. VII. Если $i < n$, то перейти к шагу VIII; иначе прекратить вычисления.

VIII. Вычислить $\| a^{i+1} G_i \|_1$, если $\| a^{i+1} G_i \|_1 > \varepsilon$, то перейти к шагу XV; иначе перейти к шагу IX.

IX. Вычислить вектор-столбец r^{i+1} размерности m

$$r^{i+1} = Q_i (a^{i+1})^T (1 + a^{i+1} Q_i (a^{i+1})^T)^{-1}.$$

X. Вычислить матрицу S_i размера $m \times i$

$$S_i = C_i^{\#} - r^{i+1} a^{i+1} C_i^{\#}.$$

XI. Составить матрицу $C_{i+1}^{\#}$ размера $m \times (i+1)$, состоящую из матрицы S_i и вектор-столбца r^{i+1}

$$C_{i+1}^{\#} = (S_i \vdots r^{i+1}).$$

XII. Положить $G_{i+1} = G_i$.

XIII. Вычислить матрицу Q_{i+1} размера $m \times m$ по формуле

$$Q_{i+1} = (I - r^{i+1}a^{i+1}) Q_i.$$

XIV. Положить $i = i + 1$ и перейти к шагу VII.

XV. Вычислить вектор-столбец r^{i+1} размера m

$$r^{i+1} = G_i (a^{i+1})^T (a^{i+1} G_i (a^{i+1})^T)^{-1}.$$

XVI. Вычислить матрицу S_i размера $m \times i$

$$S_i = (I - r^{i+1}a^{i+1}) C_i^{\#}.$$

XVII. Составить матрицу $C_{i+1}^{\#}$ размера $m \times (i+1)$, состоящую из матрицы S_i и вектор-столбца r^{i+1}

$$C_{i+1}^{\#} = (S_i \vdots r^{i+1}).$$

XVIII. Вычислить матрицу G_{i+1} размера $m \times m$

$$G_{i+1} = (I - r^{i+1}a^{i+1}) G_i.$$

XIX. Вычислить матрицу Q_{i+1} размера $m \times m$ по формуле

$$Q_{i+1} = (I - r^{i+1}a^{i+1}) Q_i (I - r^{i+1}a^{i+1})^T + r^{i+1} (r^{i+1})^T.$$

XX. Вычислить величину

$$\delta' = | \operatorname{tr} G_{i+1} - \operatorname{rang} G_{i+1} |,$$

где $\operatorname{tr} G_{i+1}$ — след матрицы G_{i+1}; $\operatorname{rang} G_{i+1}$ — ранг матрицы G_{i+1}.

XXI. Если $\delta' \leqslant \delta$, то положить $i = i + 1$ и перейти к шагу VII; иначе перейти к шагу XXII.

(Шаги XXII—XXXVII алгоритма уточняют полученные выше матрицы $C_{i+1}^{\#}$, G_{i+1}, Q_{i+1} на основании матрицы $C_{l\pm 1}$).

XXII. Положить $\overline{Q}_0^1 = Q_{i+1}$.

XXIII. Положить $j = 1$.

XXIV. Вычислить матрицу $(\overline{C}_1^j)^{\#}$ размера $m \times 1$

$$(\overline{C}_1^j)^{\#} = \overline{Q}_0^j (a^1)^T (\rho^{-2}\mu^2 + a^1 \overline{Q}_0^j (a^1)^T)^{-1}.$$

XXV. Положить $l = 0$.

XXVI. Вычислить вектор-столбец $\bar{r}^{l+1}$ размера m

$$\bar{r}^{l+1} = \overline{Q}_l^j (a^{l+1})^T (\rho^{-2}\mu^2 + a^{l+1} \overline{Q}_l^j (a^{l+1})^T)^{-1}.$$

XXVII. Вычислить матрицу $\overline{Q}_{l+1}^j$ размера $m \times m$

$$\overline{Q}_{l+1}^j = (I - \bar{r}^{l+1} a_{l+1}^j) \overline{Q}_l^j.$$

XXVIII. Если $l = 0$, то перейти к шагу XXXI; иначе перейти к шагу XXIX.

XXIX. Вычислить матрицу S_l^j размера $m \times l$

$$S_l^j = (I - \bar{r}^{l+1} a^{l+1})(\bar{C}_l^j)^{\#}.$$

XXX. Составить матрицу $(\bar{C}_{l+1}^j)^{\#}$ размера $m \times (l+1)$, состоящую из матрицы S_l^j и вектор-столбца $\bar{r}^{l+1}$

$$(\bar{C}_{l+1}^j)^{\#} = (S_l^j \vdots \bar{r}^{l+1}).$$

XXXI. Если $l < i$, то положить $l = l + 1$ и перейти к шагу XXVI; иначе перейти к шагу XXXII.

XXXII. Вычислить матрицу $(C_{i+1}^j)^{\#}$ размера $m \times (i+1)$

$$(C_{i+1}^j)^{\#} = (\rho + \rho^{-2}\mu^{-2})(\bar{C}_{i+1}^j)^{\#}.$$

XXXIII. Вычислить матрицу Q_{i+1}^j размера $m \times m$

$$Q_{i+1}^j = (C_{i+1}^j)^{\#}((C_{i+1}^j)^{\#})^T.$$

XXXIV. Вычислить матрицу G_{i+1}^j размера $m \times m$

$$G_{i+1}^j = I - (C_{i+1}^j)^{\#}(C_{i+1}).$$

XXXV. Вычислить величину

$$\delta'' = |\operatorname{tr} G_{i+1}^j - \operatorname{rang} G_{i+1}|.$$

Если $\delta'' \leqslant \delta$, то перейти к шагу XXXVI; иначе перейти к шагу XXXVII.

XXXVI. Положить $C_{i+1}^{\#} = (C_{i+1}^j)^{\#}$, $G_{i+1} = G_{i+1}^j$, $Q_{i+1} = Q_{i+1}^j$, $i = i + 1$ и перейти к шагу VII.

XXXVII. Положить $\bar{Q}_0^{j+1} = Q_{i+1}^j$, $j = j + 1$ и перейти к шагу XXIV.

Библиографические указания. При написании параграфа использовались работы [246, 247, 249]. Дополнительные сведения о псевдообратных матрицах, способах псевдообращения матриц, методах псевдообратных операторов можно получить в работах [244, 245, 246, 247, 248, 249, 252, 61].

2.7. Методы минимизации вдоль собственных векторов матрицы, близкой к матрице Гессе

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной непрерывно дифференцируемой функции $f_0 : R^n \to R^1$.

Предположение 1. (*i*) — минимизируемая функция $f_0(x)$ трижды непрерывно дифференцируема.

На k-й итерации приводимых здесь алгоритмов вычисляются векторы $h^{1,k}, h^{2,k}, \ldots, h^{n,k}$, представляющие собой систему, близкую к сопряженной по отношению к матрице вторых производных $\nabla_{xx}^2 f_0(x^k)$, и движение к следующему приближению x^{k+1} осуществляется в направлении этих векторов. Шаговые множители вычис-

ляются путем приближенного решения задачи одномерной минимизации. Основной вариант алгоритма обладает квадратичной, а ускоренный — кубической скоростью сходимости.

1. Основной вариант

Алгоритм 1

Начало I. Выбрать произвольное начальное приближение $x^1 \in R^n$.

II. Выбрать произвольную систему ортогональных векторов

$$h^{i,0}, \quad i = 1, \ldots, n, \quad (\|h^{i,0}\| = 1, \quad i = 1, \ldots, n).$$

III. Выбрать константы $\bar{\varepsilon}_0 > 0$, $\tau_0 > 0$, $\sigma \in \left(0, \frac{1}{2}\right)$, $\gamma > 1$, $\mu > 1$.

IV. Задать последовательности положительных чисел

$$\{\varepsilon_k\}_{k=1}^{\infty}; \quad \{\beta_k\}_{k=1}^{\infty}; \quad \{\alpha_k\}_{k=1}^{\infty},$$

сходящиеся к нулю.

V. Определить функции $\Delta : R^1_+ \times R^n \times R^n \to R^1$; $\delta : R^1_+ \times R^n \times R^n \to R^1$; $\theta : R^n \times R^1_+ \times R^1_+ \times R^n \to R^1$, соответственно, по формулам

$$\Delta(\varepsilon, x, h) = -\varepsilon^{-1}(f_0(x + \varepsilon h) - f_0(x));$$

$$\delta(\varepsilon, x, h) = f_0(x + \varepsilon\Delta(\varepsilon, x, h)h) - f_0(x);$$

$$\theta(x, \varepsilon, \nu, h) = f_0(x + \nu\Delta(\varepsilon, x, h)h) - f_0(x),$$

где $\varepsilon \in R^1_+$; $\nu \in R^1_+$; $x \in R^n$; $h \in R^n$; $\|h\| = 1$.

VI. Положить $k = 1$.

Основной цикл. VII. Положить $x^{0,k} = x^k$.

VIII. Найти симметричную матрицу $B_k(H_{k-1})$, элементы которой $b^k_{ij} = b^k_{ji}$

$$b^k_{ij} = \frac{1}{\alpha_k^2}[f_0(x^k + \alpha_k(h^{i,k-1} + h^{j,k-1})) - f_0(x^k + \alpha_k h^{i,k-1}) -$$

$$- f_0(x^k + \alpha_k h^{j,k-1}) + f_0(x^k)], \quad (i = 1, \ldots, n; \quad j = 1, \ldots, n),$$

здесь $H_{k-1} = (h^{1,k-1}, h^{2,k-1}, \ldots, h^{n,k-1})$.

IX. Применяя к симметричной матрице $B_k(H_{k-1})$ метод вращений Якоби для вычисления собственных значений и собственных векторов [360, с. 182], найти такую ортогональную матрицу A_k, что сумма квадратов внедиагональных элементов матрицы $A_k^T B_k(H_{k-1}) A_k$ меньше числа β_k^2. Положить $H_k = H_{k-1}A_k$ и обозначить через $\lambda_{i,k}$, $i = 1, \ldots, n$, собственные значения матрицы $B_k(H_{k-1})$

(В [321] предлагается также другой эффективный способ вычисления матрицы H_k, основанный на преобразовании Хаусхолдера и QL — алгоритме для вычисления собственных векторов и собствен

ных значений симметричной матрицы [360 с. 190 и 203]. В начале строится такая матрица U_k, что матрица $W_k = U_k^T B_k (H_{k-1}) U_k$ имеет трехдиагональную форму, а после с помощью алгоритма QL вычисляется матрица V_k, столбцы которой являются собственными векторами матрицы W_k, и полагают $H_k = H_{k-1} U_k V_k$).

В качестве чисел λ_{ik}, $i = 1, \ldots, n$, берут элементы главной диагонали матрицы $A_k^T B_k (H_{k-1}) A_k$.

X. Положить $i = 0$.

XI. Вычислить числа $\varepsilon_{i,k}$ и $\nu_{i,k}$, при которых выполняются следующие два неравенства

$$\theta(x^{i,k}, \varepsilon_{i,k}, \nu_{i,k}, h^{i+1,k}) \leqslant \sigma\nu_{i,k}\delta(\varepsilon_{i,k}, x^{i,k}, h^{i+1,k})/\varepsilon_{i,k};$$

$$\theta(x^{i,k}, \varepsilon_{i,k}, \nu_{i,k}, h^{i+1,k}) \leqslant -\gamma\varepsilon_{i,k}$$

и одно из неравенств

$$\theta(x^{i,k}, \varepsilon_{i,k}, \nu_{i,k}, h^{i+1,k}) \geqslant \nu_{i,k}(1-\sigma)\,\delta(\varepsilon_{i,k}, x^{i,k}, h^{i+1,k})/\varepsilon_{i,k},$$

или $\nu_{i,k} \geqslant \mu$. (Вычисление величины $\nu_{i,k}$ $(i = 0, 1, \ldots, n-1)$ рекомендуется начинать со значения, равного $(\lambda_{i+1,k})^{-1}$).

XII. Если $\varepsilon_{i,k} \geqslant \varepsilon_k$, то положить $\rho_{i,k} = \nu_{i,k}\,\Delta(\varepsilon_{i,k}, x^{i,k}, h^{i+1,k})$ и перейти к шагу XIII, если $\varepsilon_{i,k} < \varepsilon_k$, то положить $\rho_{i,k} = 0$ и перейти к шагу XIII.

XIII. Положить $x^{i+1,k} = x^{i,k} + \rho_{i,k}h^{i+1,k}$ и перейти к шагу XIV.

XIV. Если $i + 1 = n$, то положить $x^{k+1} = x^{n,k}$ и перейти к шагу XV; иначе положить $i = i + 1$ и перейти к шагу XI.

XV. Вычислить значение

$$\tau_k = \min\{\tau_0, [\max_{0\leqslant i\leqslant n-1} |\rho_{i,k-1}|]^2\}.$$

XVI. Положить $k = k + 1$ и перейти к шагу VII.

Теорема 1. *Пусть выполняется предположение 1 и (ii) — множество $\{x \mid f_0(x) \leqslant f_0(x^1), x \in \boldsymbol{R}^n\}$ ограничено; (iii) — матрица вторых производных $\nabla_{xx}^2 f_0(x)$ удовлетворяет неравенствам*

$$\gamma_1\|y\|^2 \leqslant (\nabla_{xx}^2 f_0(x)y, y) \leqslant \gamma_2\|y\|^2, \quad 0 < \gamma_1 \leqslant \gamma_2 < \infty,$$

$$\forall x, y \in \boldsymbol{R}^n; \quad (iv)\ \beta_k \leqslant \alpha_k \leqslant \tau_k, \quad k = 1, 2, \ldots .$$

Тогда, если в алгоритме 1 вычисление величины $\nu_{i,k}$ $(i = 0, \ldots, n-1)$ на шаге XI начать со значения $(\lambda_{i+1,k})^{-1}$, то последовательность $\{x^k\}_{k=1}^{\infty}$, порожденная алгоритмом 1, сходится к единственному решению x^* с квадратичной скоростью

$$\|x^{k+1} - x^*\| \leqslant \mathrm{const}\,\|x^k - x^*\|^2, \quad \mathrm{const} > 0.$$

Замечание 1. Если в алгоритме 1 матрицу H_k, состоящую из столбцов $h^{1,k}, \ldots, h^{n,k}$, вычислять по правилам

$$h^{1,k} = e^1;$$

$$h^{i+1,k} = \tilde{h}^{i+1,k}/\|\tilde{h}^{i+1,k}\|, \quad i = 1, 2, \ldots, n-1,$$

где

$$\tilde{h}^{i+1,k} = e^{i+1} - \sum_{j=1}^{i} \gamma_{ji}^{(k)} h^{j,k}, \quad i = 1, 2, \ldots, n-1;$$

$$e^i \ (i = 1, \ \ldots, \ n) \text{ — } i\text{-й орт};$$

$$\gamma_{ji}^{(k)} = \begin{cases} \dfrac{\varphi_{\alpha_k}(x^k, e^{i+1}, h^{j,k})}{\varphi_{\alpha_k}(x^k, h^{j,k}, h^{j,k})}, & \text{если } \varphi_{\alpha_k}(x^k, h^{j,k}, h^{j,k}) \geqslant \bar{\varepsilon}_0; \\ 0, & \text{если } \varphi_{\alpha_k}(x^k, h^{j,k}, h^{j,k}) < \bar{\varepsilon}_0, \end{cases}$$

$\bar{\varepsilon}_0 > 0$ — константа алгоритма; $\varphi_\alpha : \boldsymbol{R}^n \times \boldsymbol{R}^n \times \boldsymbol{R}^n \to \boldsymbol{R}^1$,

$$\varphi_\alpha(x, e, h) = \alpha^{-2} [f_0(x + \alpha(e + h)) - f_0(x + \alpha e) - f_0(x + \alpha h) + f_0(x)], \quad x, e, h \in \boldsymbol{R}^n, \tag{2.17}$$

а через $\lambda_{i,k}$, $i = 1, \ldots, n$, обозначить числа $\varphi_{\alpha_k}(x^k, h^{i,k}, h^{i,k})$, $i = 1, \ldots$ $\ldots, n$, то для такого модифицированного алгоритма 1 теорема 1 будет справедливой.

2. Ускоренный вариант

Алгоритм 2

Начало. I—VI. Шаги I—VI такие, как и в алгоритме 1.

Основной цикл. VII—XIV. Шаги VII—XIV такие, как и в алгоритме 1.

XV. Положить $\bar{x}^k = x^{n,k}$.

XVI. Вычислить значение t_k, удовлетворяющее условиям теоремы 2.

XVII. Вычислить вектор a^k, i-я компонента которого

$$a_i^k = [f_0(\bar{x}^k + t_k h^{i,k}) - f_0(\bar{x}^k - t_k h^{i,k})]/(2t_k), \quad i = 1, \ \ldots, \ n.$$

XVIII. Вычислить вектор

$$\bar{y} = \bar{x}^k - H_k \Lambda_k^{-1} a^k,$$

где $\Lambda_k = A_k^T B_k (H_{k-1}) A_k$ — диагональная матрица, полученная при вычислении собственных векторов матрицы $B_k (H_{k-1})$.

(Для вычисления $\bar{y}$ нет необходимости вычислять обратную матрицу к Λ_k. Достаточно обратить лишь диагональные элементы, так как оценка, полученная для недиагональных элементов, обеспечивает кубическую скорость сходимости).

XIX. Если выполняется неравенство $f_0(\bar{y}) < f_0(\bar{x}^k)$, то положить $x^{k+1} = \bar{y}$ и перейти к шагу XX; если $f_0(\bar{y}) \geqslant f_0(\bar{x}^k)$, то положить $x^{k+1} = \bar{x}^k$ и перейти к шагу XX.

XX. Положить $k = k + 1$ и перейти к шагу VII.

Теорема 2. *Если выполняется предположение 1, условия (ii), (iii) теоремы 1 и*

$$0 < t_k \leqslant [\max_{0 \leqslant i \leqslant n-1} |\rho_{i,k}|]^3, \quad \beta_k \leqslant \alpha_k \leqslant \| x^k - \bar{x}^{k-1} \|^3.$$

то для скорости сходимости последовательности $\{x^k\}_{k=1}^{\infty}$, *порожденной алгоритмом 2, справедлива оценка*

$$\| x^{k+1} - x^* \| \leqslant \text{const} \| x^k - x^* \|^3, \quad \text{const} > 0.$$

Замечание 2. Если в алгоритме 1 матрицу H_k вычислять согласно замечанию 1, то в ускоренном варианте для такого модифицированного алгоритма вектор $a^k = (a_1^k, \ldots, a_n^k)$ следует вычислять по формуле

$$a_i^k = [f_0(\bar{x}^k + t_k e^i) - f_0(\bar{x}^k - t_k e^i)]/(2t_k), \quad i = 1, \ldots, n,$$

а вектор $\bar{y}$ по формуле

$$\bar{y} = \bar{x}^k - \sum_{i=1}^{n} \frac{(a^k, h^{i,k})}{\varphi_{\alpha_k}(x^k, h^{i,k}, h^{i,k})} h^{i,k},$$

где φ_α определяется по формуле (2.17).

Библиографические указания. Параграф написан на основании работ [322, 323, 324].

2.8. Итеративные стохастические методы, использующие аналог функции Ляпунова

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Приводимый ниже метод основан на введении аналога функции Ляпунова $r_1(x)$, т. е. функции $r_1(x)$, удовлетворяющей условию 1.

Условие 1. Функция $r_1(x)$ неотрицательна, $\inf r_1(x) = 0$, дифференцируема, а ее градиент удовлетворяет условию Гельдера

$$\| \nabla r_1(x) - \nabla r_1(y) \| \leqslant \chi \| x - y \|^t, \quad 0 < t \leqslant 1, \quad 0 < \chi < \infty.$$

От $r_1(x)$ не требуется существования точек минимума. Поэтому в качестве $r_1(x)$ можно брать $r_1(x) = f_0(x) - \inf f_0(x)$ даже в том случае, когда минимизируемая функция f_0 не обязательно имеет точку минимума. Часто в качестве $r_1(x)$ берется $r_1(x) = d_1(x, X^*)$, где $d_1(x, X^*)$ — расстояние от точки x до множества решений X^* задачи 1.

На k-й итерации в качестве вектора, определяющего направление движения к следующему приближению x^{k+1}, выбирается реализация случайного вектора $\bar{\xi}^k$, удовлетворяющего условиям 2 и 3.

Условие 2. Выполняется следующее обобщение условия псевдоградиентности относительно $r_1(x)$:

$$(\nabla r_1(x^k), E\bar{\xi}^k) \geqslant \theta_k r_1(x^k) - \beta_k, \ \theta_k > 0, \ \beta_k \geqslant 0.$$

Условие 3. (*i*) — распределение случайного вектора $\bar{\xi}^k$ зависит только от k и x^k, т. е. $\bar{\xi}^k = \bar{\xi}^k(x^k)$; (*ii*) —

$$\bar{\xi}^k(x) = h^k(x) + \zeta^k(x), \quad E\zeta^k(x) = 0,$$

где помехи $\zeta^k(x)$ взаимно независимы, а $h^k(x) = E\bar{\xi}^k(x)$ — детерминирована составляющая процесса; (*iii*) —

$$E\|\bar{\xi}^k\|^{t+1} \leqslant \sigma_k^{t+1} + \tau_k r_1(x^k), \quad \sigma_k \geqslant 0, \quad \tau_k \geqslant 0.$$

1. Основной алгоритм

Алгоритм 1.

Начало. I. Выбрать функцию $r_1(x)$, удовлетворяющую условию 1.

II. Выбрать начальное приближение $x^0 \in R^n$, удовлетворяющее условию $Er_1(x^0) < \infty$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить шаговый множитель ρ_k.

V. Вычислить реализацию ξ^k случайного вектора $\bar{\xi}^k$, удовлетворяющего условиям 2 и 3.

VI. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \xi^k.$$

VII. Положить $k = k + 1$ и перейти к шагу IV.

2. Сходимость алгоритма в среднем

Теорема 2. Если выполняются условия:

(*i*) —

$$0 \leqslant \nu_k \leqslant 1, \ k = 0, 1, \ldots, \ \sum_{k=0}^{\infty} \nu_k = \infty,$$

где $\nu_k = \rho_k(\theta_k - \chi \rho_k^t \tau_k/(t+1))$;

(*ii*)— $$\overline{\lim_{k\to\infty}} \mu_k \leqslant \mu, \quad \mu \geqslant 0,$$

где

$$\mu_k = \left(\rho_k \beta_k + \frac{\chi}{t+1} \rho_k^{t+1} \sigma^{t+1}\right) / \nu_k,$$

то для последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, справедливо*

$$\overline{\lim_{k\to\infty}} Er_1(x^k) \leqslant \mu.$$

Если при этом $\mu_k \leqslant \mu$ *для всех* k, *то*

$$Er_1(x^k) \leqslant Er_1(x^0) \prod_{i=0}^{k-1} (1 - \nu_i) + \mu \left(1 - \prod_{i=0}^{k-1} (1 - \nu_i)\right).$$

Теорема 2'. Если выполняются условия:

(*i*) —

$$t = 1; \quad \theta_k = \theta, \quad k = 0, 1, \ldots; \quad \beta_k = 0, \quad k = 0, 1, \ldots;$$
$$\sigma_k = \sigma; \quad k = 0, 1, \ldots; \quad \tau_k = \tau, \quad k = 0, 1, \ldots;$$

(ii) — шаговый множитель ρ_k в алгоритме 1 постоянный, причем

$$\rho_k = \rho, \quad 0 < \rho < 2\theta/\chi\tau, \quad k = 0, 1, \ldots,$$

то

$$\overline{\lim_{k\to\infty}} E r_1(x^k) \leqslant \chi\sigma^2\rho/(2\theta - \chi\tau\rho)$$

и

$$E r_1(x^k) \leqslant E r_1(x^0)(q_1)^k + \chi\sigma^2\rho(1 - (q_1)^k)/(2\theta - \chi\tau\rho),$$

где $q_1 = 1 - \rho(\theta - \chi\tau\rho/2) < 1$.

Теорема 2 гарантирует сходимость алгоритма 1 в среднем в области малых значений функции $r_1(x)$ и дает оценку скорости сходимости. Теорема 2′ гарантирует сходимость в среднем алгоритма 1 со скоростью геометрической прогрессии в области, где $r_1(x) \leqslant \gamma$, причем $\gamma = \chi\sigma^2\rho/(2\theta - \chi\tau\rho)$ тем меньше, чем меньше ρ и чем меньше аддитивные помехи (т. е. σ^2). В ситуации с мультипликативными помехами (т. е. когда $\sigma = 0$) можно выбирать постоянный шаговый множитель $\rho_k \equiv \rho$, и если $0 < \rho < 2\theta/\chi\tau$, то имеет место сходимость в среднем со скоростью геометрической прогрессии.

Теорема 2″. *Если выполняется условие (i) теоремы 2 и* $\lim\limits_{k\to\infty} \mu_k = 0$, *то последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, такова, что*

$$\lim_{k\to\infty} E r_1(x^k) = 0.$$

Если, кроме того,

(i) —

$$\overline{\lim_{k\to\infty}} \gamma_k \leqslant \bar{\gamma} < 1,$$

где

$$\gamma_k = (\mu_k/\mu_{k+1} - 1)/\nu_k, \quad k = 0, 1, \ldots,$$

то

$$E r_1(x^k) \leqslant \mu_k/(1 - \bar{\gamma}) + o(\mu_k);$$

(ii) —

$$\gamma_k \leqslant \bar{\gamma} < 1 \text{ для всех } k, \text{ то}$$

$$E r_1(x^k) \leqslant \mu_k \left(\frac{1}{1-\bar{\gamma}} + \left(\frac{E r_1(x^0)}{\mu_0} - \frac{1}{1-\bar{\gamma}} \right) \prod_{i=0}^{k-1} (1 - (1 - \bar{\gamma})\nu_i) \right);$$

(iii) —

$$\underline{\lim_{k\to\infty}} \gamma'_k \geqslant \bar{\gamma} > 1,$$

где

$$\gamma'_k = (1 - \mu_{k+1}/\mu_k)/\nu_{k+1}, \quad k = 0, 1, \ldots,$$

то

$$Er_1(x^k) = O\left(\prod_{i=0}^{k-1}(1-\nu_i)\right);$$

(iv) —

$$\gamma_k' \geqslant \bar\gamma > 1 \text{ для всех } k, \text{ то}$$

$$Er_1(x^k) \leqslant (Er_1(x^0) + \mu_0/(\bar\gamma - 1)) \prod_{i=0}^{k-1}(1-\nu_i).$$

Теорема 2‴. *Если выполняется условие (i) теоремы 2′ и условия*

(i) —

$$0 \leqslant \rho_k \leqslant 2(\theta - \varepsilon)/\chi\tau, \quad 0 < \varepsilon < \theta, \quad \sum_{k=0}^{\infty}\rho_k = \infty;$$

(ii) —

$$\rho_k \to 0 \quad \text{при} \quad k \to \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, такова, что

$$Er_1(x^k) \to 0 \quad \text{при} \quad k \to \infty.$$

Если при этом:

(iii) — *$k\rho_k$ монотонно возрастает, $k\rho_k \to \rho > \theta^{-1}$ (возможно, $\rho = \infty$), то*

$$Er_1(x^k) \leqslant (\chi\sigma^2/2(\theta - \rho^{-1}))\rho_k + o(\rho_k);$$

(iv) — *$k\rho_k$ монотонно убывает, $k\rho_k \to \rho < \theta^{-1}$ (возможно, $\rho = 0$), то*

$$Er_1(x^k) = O\left(e^{-\theta\sum_{i=0}^{k-1}\nu_i}\right),$$

где e — *основание натурального логарифма.*

Теорема 2″ является дискретным стохастическим аналогом теорем о сходимости по функционалу в детерминированных методах оптимизации и в ней даются оценки скорости сходимости как асимптотические (утверждения (*i*), (*iii*)), так и справедливые при всех k (утверждения (*ii*), (*iv*)). В теореме 2‴ утверждается, что при наличии аддитивной помехи для сходимости в среднем достаточно условий

$$\sum_{k=0}^{\infty}\rho_k = \infty; \quad \rho_k \to 0 \quad \text{при} \quad k \to \infty.$$

Скорость сходимости $Er_1(x^k)$ зависит от скорости убывания ρ_k, причем оценка для $Er_1(x^k)$ не лучше, чем $O(1/k)$.

Замечание 2. Если функция $r_1(x)$ имеет единственную точку минимума x^*, причем выполняется условие

$$r_1(x) \geqslant \lambda\|x - x^*\|^2, \quad \lambda > 0$$

(например, если $r_1(x)$ сильновыпукла), то из сходимости $Er_1(x^k) \to 0$ следует средняя квадратическая сходимость x^k к x^*, т. е.

$$E(\|x^k - x^*\|^2) \to 0 \quad \text{при} \quad k \to \infty.$$

3. Сходимость алгоритма почти наверное

Теорема 3. *Если выполняется условие* (*i*) *теоремы 2 и*

$$\sum_{k=0}^{\infty} \mu_k \nu_k < \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, такова, что*

$$r_1(x^k) \xrightarrow{\text{п.н.}} 0 \ (\text{т. е. } P\{r_1(x^k) \to 0\} = 1).$$

При этом для всех $\varepsilon_0 > 0$, $k_0 \geqslant 0$,

$$P\{r_1(x^k) \leqslant \varepsilon_0,\ k \geqslant k_0\} \geqslant 1 - \left(Er_1(x^{k_0}) + \sum_{k=k_0}^{\infty} \mu_k \nu_k\right)/\varepsilon_0.$$

Если, кроме того, $\underline{\lim}_{k\to\infty} \gamma_k' \geqslant \bar{\gamma} > 1$, *то для всякого* $l_1 > 0$ *найдется* $l_2 = l_2(l_1, Er_1(x^0))$ *такое, что*

$$P\left\{r_1(x^k) \leqslant (l_1 + l_2) \prod_{i=0}^{k-1} (1 - \nu_i) \text{ для всех } k\right\} \geqslant 1 - l_2/l_1,$$

а если $\gamma_k' \geqslant \bar{\gamma} > 1$ *для всех* k, *то*

$$P\left\{r_1(x^k) \leqslant (l_1 + Er_1(x^0) + \mu_0/(\bar{\gamma} - 1)) \prod_{i=0}^{k-1} (1 - \nu_i) \text{ для всех } k\right\} \geqslant$$
$$\geqslant 1 - Er_1(x^0)/l_1 - \mu_0/l_1(\bar{\gamma} - 1).$$

Теорема 3'. *Если выполняются условие* (*i*) *теоремы 2', условие* (*i*) *теоремы 2''' и* $\sigma^2 = 0$, *либо* $\sum_{k=0}^{\infty} \rho_k^2 < \infty$, *то* $r_1(x^k) \xrightarrow{\text{п.н}} 0$. *Если, кроме того,* $k\rho_k$ *монотонно убывает и* $k\rho_k \to \rho < \theta^{-1}$, *то для всякого* $l_1 > 0$ *найдется такое* $l_2 = l_2(l_1, Er_1(x^0))$, *что*

$$P\left\{r_1(x^k) \leqslant (l_1 + l_2)\, e^{-\theta \sum_{i=0}^{k-1} \nu_i} \text{ для всех } k\right\} \geqslant 1 - l_2/l_1.$$

В частности, если $\rho_k = \rho/k$, $k \geqslant 1$, $\rho\theta < 1$, *то для всякого* $l_1 > 0$

$$P\{r_1(x^k) \leqslant l_1/k^{\rho\theta} \text{ для всех } k \geqslant 1\} \geqslant$$
$$\geqslant 1 - Er_1(x^1)/l_1 - \chi\sigma^2\rho^2\theta/(l_1(2\theta - \chi\rho\tau)(1 - \rho\theta)).$$

Замечание 3. Если функция $r_1(x)$ имеет единственную точку минимума x^* и $r_1(x^*) = 0$, $\inf_{\|x - x^*\| \geqslant \varepsilon} r_1(x) > 0$ при некотором $\varepsilon > 0$, то из сходимости $r_1(x^k) \xrightarrow{\text{п.н}} 0$ следует, что $x^k \xrightarrow{\text{п.н}} x^*$.

4. Модификации алгоритма

Если условие 2 в алгоритме 1 заменить на 2′

$$(\nabla r_1(x^k),\ E\xi^k) \geqslant -\beta_k, \quad \beta_k \geqslant 0, \quad \beta_k \to 0,$$

то при

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \mu_k \nu_k < \infty, \quad \sum_{k=0}^{\infty} \rho_k^{t+1} \tau_k < \infty,$$

последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная модифицированным алгоритмом, такова, что $r_1(x^k)$ сходится почти наверное к пределу, а

$$\varliminf_{k\to\infty} (\nabla r_1(x^k),\ E\xi^k) \overset{\text{п.н}}{=} 0.$$

Если на шаге VI алгоритма 1 следующее приближение x^{k+1} вычислять по формуле

$$x^{k+1} = \pi_X(x^k - \rho_k \xi^k),$$

где π_X — оператор проектирования на некоторое множество X, то для такого алгоритма все предыдущие теоремы остаются справедливыми.

Библиографические указания. Параграф написан на основании работы [296]. Результаты типа теоремы 2″ о сходимости алгоритма 1 (но с менее полными оценками скорости сходимости) приведены в [3, 263]. Утверждения о сходимости почти наверное в условиях теоремы 3′ приведены в [3, 263], [436]. Вопросы локальной сходимости рассматривались в [213, 513]. В [297] для линейного случая приведены более точные оценки сходимости алгоритма 1.

2.9. Стохастические квазиградиентные методы

1. Общий стохастический квазиградиентный метод для детерминированных задач

Задача 1. Найти $\arg\min_x f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположения 1. (*i*) Функция f_0 непрерывно дифференцируема; (*ii*) градиент функции f_0 удовлетворяет условию Липшица, т. е. для любых точек x, y

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma \|x - y\|, \quad 0 < \gamma < \infty.$$

В стохастических квазиградиентных методах последующие приближения к искомому решению находят в направлении стохастического квазиградиента — случайного вектора ξ^k, условное математическое ожидание которого в некотором смысле близко к градиенту (или обобщенному градиенту) рассматриваемой функции, т. е.

$$E(\xi^k / x^0, \ldots, x^k) = \alpha_k \hat{\nabla} f_0(x^k) + b^k, \quad b^k \to 0, \quad \alpha_k \to \text{const} > 0.$$

Алгоритм 1

Начало. I. Выбрать произвольную начальную точку $x^0 \in R^n$ и числа ρ_0, γ_0.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализацию ξ^k случайного вектора ξ, условное математическое ожидание которого удовлетворяет равенству

$$E(\xi/x^0, \ldots, x^k) = \alpha_k \nabla f_0(x^k) + b^k,$$

где α_k — неотрицательная случайная величина и b^k — случайный вектор, измеримые относительно σ-подалгебры $\mathcal{B}_k$, индуцированной семейством случайных величин $(x^0, \ldots, x^k)$; $\nabla f_0(x^k)$ — градиент функции f_0 в точке x^k.

IV. Вычислить вектор $x^{k+1} = x^k - \rho_k \gamma_k \xi^k$.

V. Вычислить шаговый ρ_{k+1} и нормирующий γ_{k+1} множители.

IV. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть: выполнены предположения 1 и, кроме того, для каждого числа $\delta < \infty$ существует число $C_\delta < \infty$ такое, что $\| \nabla f_0(x) \| \leqslant C_\delta$ при $f_0(x) \leqslant \delta$; $\eta_k(\omega)$ — случайная величина, измеримая относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами $(x^0, \ldots, x^k)$, такая, что для любого $\delta < \infty$ и некоторого числа c_δ*

$$E(\|\xi^k\|^2/x^0, \ldots, x^k) \leqslant \eta_k^2 \leqslant c_\delta$$

при $f_0(x^s) \leqslant \delta$, $s = 0, 1, \ldots, k$; нормирующий множитель γ_k удовлетворяет условию $0 < \gamma \leqslant \gamma_k \eta_k \leqslant r_k < \infty$; кроме того, величины ρ_k, r_k, измеримые относительно σ-подалгебры $\mathcal{B}_k$, такие, что $\rho_k \geqslant 0$, $\alpha_k \geqslant 0$, $\| b^k \|/\alpha_k \to 0$ равномерно с вероятностью 1 и

$$\sum_{k=0}^{\infty} E(\rho_k \| b^k \| + \rho_k^2 r_k^2) < \infty;$$

$$\sum_{k=0}^{\infty} \rho_k \alpha_k \overset{\text{п.н}}{=\!=} \infty.$$

Тогда последовательность $\{x^k(\omega)\}_{k=0}^{\infty}$, порожденная алгоритмом 1, такова, что почти для каждого ω последовательность $\{f_0(x^k(\omega))\}_{k=0}^{\infty}$ сходится, и $\| \nabla f_0(x^{k_s}(\omega)) \| \to 0$ почти наверное при $s \to \infty$ для некоторой подпоследовательности

$$\{x^{k_s}(\omega)\}_{s=0}^{\infty}.$$

Теорема 1'. *Пусть выполнены предположения 1 и, кроме того, для каждого числа $\delta < \infty$ существует число $C_\delta < \infty$ такое, что $\| \nabla f_0(x) \| \leqslant C_\delta$ при $f_0(x) \leqslant \delta$; $\alpha_k \equiv 1$; $d_k(\omega)$ — случайная величина, измеримая относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами $(x^0, \ldots, x^k)$ такая, что для любого δ и некоторого числа $c_\delta < \infty$*

$$\sum_{j=1}^{n} D(\xi_j^k/x^0, \ldots, x^k) \leqslant d_k^2 \leqslant c_\delta$$

при $f_0(x^s) \leqslant \delta$, $s = 0, 1, \ldots, k$; *нормирующий множитель* γ_k *удовлетворяет условию* $0 < \gamma \leqslant \gamma_k d_k \leqslant r_k < \infty$; *кроме того, величины* ρ_k, r_k, *измеримые относительно σ-подалгебры* $\mathcal{B}_k$, *и такие, что* $\rho_k \geqslant 0$, $\rho_k \to 0$, $\|b^k\| \to 0$ *равномерно с вероятностью 1 и*

$$\sum_{k=0}^{\infty} E(\rho_k \|b^k\| + \rho_k^2 r_k^2) < \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty$$

с вероятностью 1. Тогда последовательность точек $\{x^k(\omega)\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, такова, что почти для каждого* ω *последовательность* $\{f_0(x^k(\omega))\}_{k=0}^{\infty}$ *сходится, причем* $\|\nabla f_0(x^{k_s}(\omega))\| \to 0$ *почти наверное при* $s \to \infty$.

Следствие. Если $f_0(x)$ унимодальна, т. е. имеет только глобальный экстремум, и $\|\nabla f_0(x)\| \neq 0$ во всех точках x, за исключением оптимальных, то из теорем 1 и 1′ следует, что последовательность $\{f_0(x^k(\omega))\}_{k=0}^{\infty}$ почти наверное сходится к глобальному экстремуму.

2. Общий стохастический квазиградиентный метод для стохастических задач

Задача 2. Найти $\arg\min_x E\varphi_0(x, \omega)$ для заданной функции $\varphi_0 : R^n \times \Omega \to R^1$.

Предположение 2. Функция $f_0(x) \triangleq E\varphi_0(x, \omega)$ дважды непрерывно дифференцируема.

Алгоритм 2

Начало. I. Выбрать: произвольное начальное приближение $x^0 \in R^n$; начальные значения нормирующего и шагового множителей ρ_0 и γ_0, начальное значение величины смещения Δ_0 (числа ρ_0, γ_0 и Δ_0 выбираем в соответствии с условиями теоремы 2).

II. Положить $k = 0$.

Основной цикл. III. Выбрать натуральное число $s_k \geqslant 1$.

IV. Найти $\{\xi^{k,s}\}_{s=1}^{s_k}$ — серию независимых наблюдений вектора $\xi = (\xi_1, \ldots, \xi_n)$ с независимыми и равномерно распределенными на $[-1, 1]$ компонентами в k-й итерации.

V. Найти $\{\omega^{k,s}\}_{s=1}^{s_k}$ — серию независимых по $k = 0, 1, \ldots$, наблюдений состояний природы ω.

VI. Вычислить вектор

$$x^{k+1} = x^k - \rho_k \gamma_k \sum_{s=1}^{s_k} \frac{\varphi_0(x^k + \Delta_k \xi^{k,s}, \omega^{k,s}) - \varphi_0(x^k, \omega^{k,s})}{\Delta_k} \xi^{k,s}.$$

VII. Вычислить шаговый множитель ρ_{k+1}, нормирующий множитель γ_{k+1} и смещение Δ_{k+1}, удовлетворяющие условиям теоремы 2.

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. Пусть выполнено предположение 2 и условия: (i) — $\mu_k(\omega)$ — случайная величина, измеримая относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами $(x^0, \ldots, x^k)$, такая, что для любого α и некоторого числа $\tau(\alpha) < \infty$

$$\sum_{j=1}^{n} D\left(\frac{\varphi_0(x^k + \Delta_k \xi^{k,s}, \omega^{k,s}) - \varphi_0(x^k, \omega^{k,s})}{\Delta_k} \xi^{k,s}/x^k\right) \leqslant \mu_k^2 \leqslant \tau(\alpha)$$

при $f_0(x^s) \leqslant \alpha$, $s = 0, 1, \ldots, k$; (ii) — нормирующий множитель γ_k удовлетворяет условию $0 < \gamma \leqslant \gamma_k \mu_k \leqslant r_k < \infty$; (iii) — величины ρ_k, r_k, Δ_k, измеримые относительно σ-подалгебры $\mathcal{B}_k$, и такие, что $\rho_k \geqslant 0$, $\rho_k \to 0$, $\Delta_k \to 0$ равномерно с вероятностью 1 и

$$\sum_{k=0}^{\infty} E(\rho_k \Delta_k + \rho_k^2 r_k^2) < \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty$$

с вероятностью 1.

Тогда последовательность точек $\{x^k(\omega)\}_{k=0}^{\infty}$, порождаемая алгоритмом 2, такова, что почти для каждого ω последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ сходится, причем $\|\nabla f_0(x^{k_s})\| \to 0$ почти наверное при $s \to \infty$ для некоторой подпоследовательности k_s.

3. Стохастический квазиградиентный метод с процедурой прерывания

Задача 3. Найти $\arg\min_x f_0(x)$ для непрерывно дифференцируемой функции $f_0 : R^n \to R^1$.

Стохастический квазиградиентный метод с процедурой прерывания основан на построении в k-й итерации вектора стохастического квазиградиента ξ^k, условное математическое ожидание которого

$$E(\xi^k/x^0, \ldots, x^k) = \nabla f_0(x^k) + b^k,$$

где b^k — вектор, измеримый относительно σ-подалгебры $\mathcal{B}_k$, индуцированной случайными величинами $(x^0, \ldots, x^k)$. Если в какой-то итерации движение по стохастическому квазиградиенту выходит за пределы сферы определенного радиуса, то происходит прерывание, и алгоритм начинает работать с произвольной точки, принадлежащей наперед заданному ограниченному замкнутому множеству.

Алгоритмы такого рода отражают те часто встречающиеся на практике ситуации, когда в процессе вычислений приходится менять либо параметры алгоритма, либо сам алгоритм.

Алгоритм 3

Начало. I. Выбрать произвольную константу α $(0 < \alpha < \infty)$ и задать произвольное замкнутое множество B, содержащееся в сфере радиуса α, т. е.

$$B \subset S \triangleq \{x \mid \|x\| \leqslant \alpha\}.$$

Выбрать произвольную начальную точку $x^0 \in B$.

II. Задать правило формирования последовательности шаговых множителей $\{\rho_k\}_{k=0}^{\infty}$.

III. Положить $k = 0$; найти ρ_0.

Основной цикл. IV. Вычислить случайный вектор ξ^k, удовлетворяющий условию

$$E(\xi^k / x^0, \ldots, x^k) = \nabla f_0(x^k) + b^k,$$

где b^k — вектор, измеримый относительно σ-подалгебры $\mathcal{B}_k$, индуцированной $(x^0, \ldots, x^k)$.

V. Вычислить вектор

$$x^{k+1} = \begin{cases} x^k - \rho_k \xi^k, & \text{если } \|x^k(\omega)\| \leqslant \alpha; \\ z^{k+1} \in B, & \text{если } \|x^k(\omega)\| > \alpha, \end{cases}$$

где z^{k+1} — произвольная точка множества B.

VI. Найти ρ_{k+1}.

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 3. *Пусть f_0 непрерывно дифференцируема функция и выполнены условия: (**i**) — существует такая постоянная δ, что*

$$\|\xi^k\| + \|\nabla f_0(x^k)\| + \|b^k\| \leqslant \delta;$$

(***ii***) —

$$\max_{x \in B} f_0(x) \leqslant \inf_{\|x\| > \alpha} f_0(x);$$

***(iii)** — величины ρ_k, измеримые относительно σ-подалгебры $\mathcal{B}_k$, удовлетворяют условиям*

$$\rho_k \geqslant 0, \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k \|b^k\| < \infty \text{ п. н.};$$

$$\sum_{k=0}^{\infty} E\rho_k^2 < \infty.$$

Тогда последовательность $\{x^k(\omega)\}_{k=0}^{\infty}$, порожденная алгоритмом 3, такова, что почти для каждого ω последовательность $\{f_0(x^k(\omega))\}_{k=0}^{\infty}$ сходится и любая предельная точка последовательности $\{x^k(\omega)\}_{k=0}^{\infty}$ принадлежит $X^* \triangleq \{x \mid \nabla f_0(x) = 0\}$ почти при каждом ω.

Замечание 3. В данной теореме, в отличие от теоремы 1, не требуется глобального условия Липшица. Имеющиеся в теореме 3 условия равномерной по k ограниченности (условие (*i*)) легко ослабить, используя нормирующий множитель γ_k. Эти условия ограниченности, однако, выполнимы, если только точки x^k изменяются в ограниченной области, а случайные помехи имеют усеченные законы распределения, т. е. принадлежат также ограниченной области.

4. Стохастический градиентный метод с постоянным шаговым множителем

Задача 4. Найти $\arg\min_x E_\omega f_0(x, \omega)$ для заданной функции $f_0 : R^n \times \Omega \to R^1$.

Предположение 4. Почти при всех ω выполняются неравенства:

$$\|\nabla f_0(x, \omega)\| \leqslant T < \infty, \quad \forall x \in \mathbf{R}^n;$$

$$\|\nabla f_0(x, \omega) - \nabla f_0(y, \omega)\| \leqslant L\|x - y\|, \quad \forall x, y \in \mathbf{R}^n;$$

$$\min_{x \in \mathbf{R}^n} \mathbf{E}_\omega f_0(x, \omega) \triangleq F^* > -\infty.$$

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \mathbf{R}^n$, шаговый множитель $\lambda > 0$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализацию ω^k случайного параметра ω.

IV. Вычислить $\nabla f_0(x^k, \omega^k)$.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \lambda \nabla f_0(x^k, \omega^k).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 4. *Пусть выполнено предположение 4, тогда для любых чисел* $\varepsilon \in (0, 1)$ *и* $\delta > 0$ *алгоритм 4 с постоянным шаговым множителем* λ,

$$0 < \lambda \leqslant \gamma\delta^2 \Big/ \left(2LT^2\left(\left[\left(1 + \frac{2LT^2(F(x^0) - F^*)}{\gamma(1-\gamma)\delta^4}\right) \times \right.\right.\right.$$

$$\left.\left.\left. \times \frac{1}{\varepsilon}\left(\frac{2LT^2}{\gamma\delta^2}\right)^2\right] + 1\right)\right), \quad 0 < \gamma < 1, \quad (F(x) \triangleq \mathbf{E}_\omega f_0(x, \omega))$$

приводит в область δ*-экстремальных решений*

$$X_\delta = \{x \mid \delta \geqslant \|\nabla \mathbf{E}_\omega f_0(x, \omega)\|\}$$

за s $(s \leqslant \varepsilon(\gamma^2\delta^4/(8n\lambda LT^4))^2)$ *итераций с вероятностью* $\mathbf{P} \geqslant 1 - \varepsilon$.

5. Стохастический градиентный метод минимизации сложных функций регрессии

Задача 5. Найти $\arg\min_{x \in \mathbf{R}^n} f_0(x, \mathbf{E}_\omega \varphi(x, \omega))$ для заданной скалярной функции $f_0 : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^1$ и заданной m-мерной функции $\varphi : \mathbf{R}^n \times \Omega \to \mathbf{R}^m$ (распределение случайного вектора ω считается неизвестным, а поэтому неизвестна функция $\mathbf{E}_\omega \varphi(x, \omega)$).

Предположение 5. (i) — существует точка минимума x^* функции $\Psi_0(x) \triangleq f_0(x, \mathbf{E}_\omega \varphi(x, \omega))$; ($ii$) — в окрестности точки (x^*, z^*) (где $z^* = \mathbf{E}_\omega \varphi(x^*, \omega)$) функция $z(x)$ (где $z(x) \triangleq \mathbf{E}_\omega \varphi(x, \omega)$) и функция $f_0(x, z)$ дифференцируемы, причем $\nabla_x f_0(x, z)$, $\nabla_z f_0(x, z)$ удовлетворяют условию Липшица, а $\nabla_x z(x)$ ограничен.

На k-й итерации алгоритма при известном x^k следующее приближение x^{k+1} находится в направлении статистической оценки анти-

градиента функции $\Psi_0(x)$, вычисляемой с помощью вспомогательного вектора y^k, приближающего значение $E_\omega \varphi(x^k, \omega)$.

Алгоритм 5

Начало. I. Выбрать произвольные начальные приближения

$$x^0 \in R^n, \; y^0 \in R^m.$$

II. Положить $k = 0$.

Основной цикл. III. Вычислить независимую реализацию ω^k случайного вектора ω.

IV. Вычислить векторы $\nabla_x f_0(x^k, y^k)$, $\nabla_z f_0(x^k, y^k)$, $\varphi(x^k, \omega^k)$ и матрицу $\nabla_x \varphi(x^k, \omega^k)$.

V. Вычислить шаговые множители ρ_k и ρ'_k, удовлетворяющие условиям теоремы 5.

VI. Вычислить следующее приближение x^{k+1} для вектора x^*

$$x^{k+1} = x^k - \rho_k [\nabla_x f_0(x^k, y^k) + (\nabla_x \varphi(x^k, \omega^k))^T \nabla_z f_0(x^k, y^k)].$$

VII. Вычислить приближение y^{k+1} для вектора $z(x^{k+1})$

$$y^{k+1} = y^k - \rho'_k [y^k - \varphi(x^k, \omega^k)].$$

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 5. *Пусть выполняются предположения 5 и (iii) — функция $\Psi_0(x)$ локально сильно выпукла в окрестности x^*, т. е. $(\nabla\Psi_0(x), x - x^*) \geqslant \nu \| x - x^* \|^2$, $\nu > 0$, для x близких к x^*; (iv) — в окрестности точки x^* помехи имеют ограниченную дисперсию:*

$$E_\omega \| \varphi(x, \omega) - z(x) \|^2 \leqslant (\sigma_1)^2;$$
$$E_\omega \| \nabla_x \varphi(x, \omega) - \nabla_x z(x) \|^2 \leqslant (\sigma_2)^2.$$

Тогда если в алгоритме 5 начальное приближение (x^0, y^0) удовлетворяет условию

$$\| x^0 - x^* \|^2 + \| y^0 - z^* \|^2 \leqslant \delta,$$

а шаговые множители ρ_k и ρ'_k такие, что

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} (\rho'_k)^2 \leqslant d_1 < \infty, \quad \rho'_k/\rho_k \geqslant \beta > 0,$$

то для достаточно большого β последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 5, удовлетворяет соотношениям

$$P\{x^k \to x^*\} \geqslant 1 - \tau, \quad \tau = c_1\delta + c_2(\sigma_2)^2 \beta d_1 + c_3(\sigma_1)^2 d_1,$$

где c_i, $i = 1, 2, 3$ — некоторые константы.

Если шаговые множители ρ_k и ρ'_k выбирать по правилам

$$\rho_k = \frac{\rho}{k+1}; \quad \rho'_k = \frac{\rho'}{k+t},$$

то для любых чисел $\tau > 0$, $c_4 > 0$ можно выбирать параметры δ, ρ, ρ', t так, что

$$P\left\{\|x^k - x^*\|^2 \leqslant \frac{c_4}{k} \text{ для всех } k\right\} \geqslant 1 - \tau.$$

Замечание 5. Из теоремы следует, что можно гарантировать сходимость алгоритма 5 с вероятностью $1 - \tau$, где τ тем меньше, чем точнее начальное приближение (x^0, y^0), чем меньше уровень помех и длина шаговых множителей. Сходимости в среднем квадратическом или с вероятностью единица в невыпуклом случае не существует, так как при наличии помех имеется ненулевая вероятность выхода последовательности $\{x^k\}_{k=0}^{\infty}$ из окрестности решения, вне которой сходимость отсутствует.

Библиографические указания. При написании параграфа использовались работы [149, 153, 29, 171, 299]. Стохастические квазиградиентные методы решения задач безусловной оптимизации исследовались также в работах [148, 150, 151, 156, 157].

2.10. Метод локальных вариаций

З а д а ч а 1. Найти $\arg\min\limits_{x \in R^n} f_0(x)$ для заданной непрерывно дифференцируемой функции $f_0 : R^n \to R^1$.

Метод локальных вариаций не требует вычисления производных минимизируемой функции и не является очевидной модификацией алгоритма, требующего таких вычислений.

Алгоритм 1

Н а ч а л о. I. Выбрать начальное приближение $x^0 \in R^n$, удовлетворяющее условиям теоремы 1; константу $\rho_0 > 0$ (рекомендуется $\rho_0 = 1$).

II. Вычислить n-мерные векторы h^i, $i = 1, 2, \ldots, 2n$,

$$h^{2j-1} = e^j, \quad j = 1, 2, \ldots, n;$$

$$h^{2j} = -e^j, \quad j = 1, 2, \ldots, n,$$

где e^j, $j = 1, \ldots, n$ — j-й столбец единичной $n \times n$-матрицы.

III. Положить $k = 0$, $x = x^0$.

IV. Вычислить $f_0(x)$.

О с н о в н о й ц и к л. V. Положить $\rho = \rho_k$.

VI. Положить $i = 1$.

VII. Вычислить $f_0(x + \rho h^i)$.

VIII. Если $f_0(x + \rho h^i) < f_0(x)$, то положить $x = x + \rho h^i$ и перейти к шагу VI; иначе перейти к шагу IX.

IX. Если $i < 2n$, то положить $i = i + 1$ и перейти к шагу VII; иначе перейти к шагу X.

X. Положить $x^{k+1} = x$, $\rho_{k+1} = \rho/2$, $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если функция f_0 непрерывно дифференцируема и на-*

чальное приближение x^0 в алгоритме 1 таково, что множество

$$X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in \boldsymbol{R}^n\}$$

ограничено, то каждая предельная точка x' последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, удовлетворяет условию $\nabla f_0(x') = 0$. Если, кроме того, множество

$$X_* \triangleq \{x \mid \nabla f_0(x) = 0, \quad x \in \boldsymbol{R}^n\}$$

состоит из конечного числа точек и каждая его точка доставляет либо локальный минимум, либо локальный максимум, то последовательность $\{x^k\}_{k=0}^{\infty}$ сходится к такой точке $\bar{x}$, что $\nabla f_0(\bar{x}) = 0$.

Метод локальных вариаций особенно эффективен, когда минимизируемая функция f_0 имеет вид

$$f_0(x) = \sum_{i=1}^{n} f_i(x_i), \quad x = (x_1, x_2, \ldots, x_n).$$

Библиографические указания. Параграф основан на работах [285, 15]. Дополнительные сведения о методе локальных вариаций можно найти в работах [385, 388, 449, 260, 261, 339].

2.11. Методы самонастраивающихся программ

Задача 1. Найти $\arg\min_{x \in \boldsymbol{R}^n} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$.

Предположение 1. Функция f_0 — непрерывно дифференцируема в $\boldsymbol{R}^n$.

В алгоритме 1 с помощью принципа самонастраивающихся программ производится соединение следующих методов: градиентного спуска, покоординатного спуска, сопряженных градиентов. На каждой итерации алгоритма определенное число раз производится градиентный спуск в подпространстве быстрых переменных и градиентный спуск в подпространстве медленных переменных. Алгоритм применим к решению задач овражного типа.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Выбрать произвольные константы $\varepsilon > 0$, $\delta > 0$ (причем $\varepsilon \ll \delta$) и четное натуральное число l.

III. Положить $m_1^0 = m_2^0 = l/2$.

IV. Положить $k^1 = 0$.

Основной цикл. V. Положить $\bar{x}^1 = x^k$.

VI. Положить $j = 1$.

VII. Вычислить вектор $h^j = (h_1^j, \ldots, h_n^j)$, i-я координата которого

$$h_i^j = \begin{cases} 0 & \text{при } \left|\frac{\partial f_0(\bar{x}^j)}{\partial x_i}\right| \leqslant \varepsilon; \\ -\frac{\partial f_0(\bar{x}^j)}{\partial x_i} & \text{в противном случае,} \end{cases}$$

здесь $i = 1, \ldots, n$.

VIII. Вычислить значение ρ_j, удовлетворяющее условию

$$\min_{\rho \geqslant 0} f_0(\bar{x}^j + \rho h^j) = f_0(\bar{x}^j + \rho_j h^j).$$

IX. Положить $\bar{x}^{j+1} = \bar{x}^j + \rho_j h^j$.

X. Если $j < m_1^k$, то положить $j = j + 1$ и перейти к шагу VII; иначе положить $\bar{x}^{k+1} = \bar{x}^{j+1}$ и перейти к шагу XI.

XI. Положить $\bar{\bar{x}}^1 = \bar{x}^{k+1}$.

XII. Положить $j = 1$.

XIII. Вычислить вектор $h^j = (h_1^j, \ldots, h_n^j)$, i-я координата которого

$$h_i^j = \begin{cases} 0 & \text{при } \left|\frac{\partial f_0(\bar{\bar{x}}^j)}{\partial x_i}\right| \geqslant \delta; \\ -\frac{\partial f_0(\bar{\bar{x}}^j)}{\partial x_i} + \theta_j h_i^{j-1} & \text{в противном случае,} \end{cases}$$

где

$$\theta_1 = 0; \ \theta_j = (\nabla f_0(\bar{\bar{x}}^j) - \nabla f_0(\bar{\bar{x}}^{j-1}), \ \nabla f_0(\bar{\bar{x}}^j)) / \| \nabla f_0(\bar{\bar{x}}^{j-1}) \| \text{ при } j > 1.$$

XIV. Вычислить значение ρ_j, удовлетворяющее условию

$$\min_{\rho \geqslant 0} f_0(\bar{\bar{x}}^j + \rho h^j) = f_0(\bar{\bar{x}}^j + \rho_j h^j).$$

XV. Положить $\bar{\bar{x}}^{j+1} = \bar{\bar{x}}^j + \rho_j h^j$.

XVI. Если $j < m_2^k$, то положить $j = j + 1$ и перейти к шагу XIII; иначе положить $x^{k+1} = \bar{\bar{x}}^{j+1}$ и перейти к шагу XVII.

XVII. Вычислить веса

$$p_1^k = 2\frac{f_0(x^k) - f_0(\bar{x}^{k+1})}{l}; \quad p_2^k = 2\frac{f_0(\bar{x}^{k+1}) - f_0(x^{k+1})}{l}.$$

XVIII. Вычислить

$$m_1^{k+1} = p_1^k \frac{l}{p_1^k + p_2^k}.$$

XIX. Вычислить

$$m_2^{k+1} = p_2^k \frac{l}{p_1^k + p_2^k}.$$

XX. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. Если градиент функции f_0 удовлетворяет условию Липшица, т. е.

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leq \gamma \|x - y\|, \quad \forall x, \ y \in R^n,$$

и множество $X_0 = \{x \mid f_0(x) \leq f_0(x^0), \ x \in R^n\}$ ограничено, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, такова, что

$$\lim_{k \to \infty} \|\nabla f_0(x^k)\| = 0.$$

Если, кроме того, f_0— дважды непрерывно дифференцируема функция, причем ее матрица вторых частных производных $\nabla^2_{xx} f_0(x)$ удовлетворяет условию

$$\beta \|y\|^2 \leq (\nabla^2_{xx} f_0(x) y, \ y) \leq \gamma \|y\|^2 \qquad (2.18)$$

при всех $x \in X_0$ и любых $y \in R^n$, то $x^k \to x^*$, $f_0(x^k) \to f_0(x^*)$, где x^*— точка минимума (единственная) функции f_0.

Для скорости сходимости справедливы следующие оценки:

$$f_0(x^k) - f_0(x^*) \leq \alpha_1 q^k; \quad \|x^k - x^*\| \leq \alpha_2 q^{k/2}, \qquad (2.19)$$

где

$$\alpha_1 = f_0(x^0) - f_0(x^*), \quad \alpha_2 = (2\alpha_1/\beta)^{1/2};$$
$$q = 1 - (\beta/2\gamma)^2.$$

Замечание 1. Шаги VIII и XIV алгоритма 1 требуют точного вычисления минимума одномерной функции, что на практике не всегда выполнимо. Можно построить модификацию алгоритма 1, требующую приближенного вычисления минимума одномерной функции. Для этого необходимо константу ε выбирать из условия $0 < \varepsilon < 1/2$, а шаги VIII и XIV алгоритма 1 заменить, соответственно, на шаги VIII' и XIV'.

VIII'. Вычислить значение ρ_j, удовлетворяющее условию

$$(1 - \varepsilon)(\nabla f_0(\bar{x}^j), \ h^j) \leq (\nabla f_0(\bar{x}^j + \rho_j h^j), \ h^j) \leq \varepsilon (\nabla f_0(\bar{x}^j), \ h^j).$$

XIV'. Вычислить значение ρ_j, удовлетворяющее условию

$$(1 - \varepsilon)(\nabla f_0(\bar{\bar{x}}^j), \ h^j) \leq (\nabla f_0(\bar{\bar{x}}^j + \rho_j h^j), \ h^j) \leq \varepsilon (\nabla f_0(\bar{\bar{x}}^j), \ h^j).$$

При выполнении условия (2.18) для скорости сходимости модифицированного таким образом алгоритма 1 справедливы оценки (2.19) при $q = 1 - 2\varepsilon(1 - \varepsilon)(\beta/\gamma)(1 + \beta/\gamma)$.

Библиографические указания. Параграф написан на основании работы [384]. Дополнительные сведения о методе самонастраивающихся программ и его модификациях можно получить из работ [381, 382, 383].

2.12. Общий метод спуска

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположение 1. (*i*) — функция f_0 выпукла и непрерывно дифференцируема в R^n.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Положить $k = 0$.

Основной цикл. III. Найти вектор h^k, удовлетворяющий условию

$$(\nabla f_0(x^k),\ h^k) \geqslant 0.$$

IV. Вычислить шаговый множитель $\rho_k > 0$ из условия

$$f_0(x^k - \rho_k h^k) = \min_{\rho \geqslant 0} f_0(x^k - \rho h^k).$$

V. Положить

$$x^{k+1} = x^k - \rho_k h^k.$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, справедлива следующая теорема.

Теорема 1. *Если выполняется предположение 1 и условия:*

(*ii*) —

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma \|x - y\|, \quad \forall\, x,\ y \in R^n, \quad (\gamma < \infty);$$

(*iii*) — *множество*

$$X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in R^n\}$$

ограничено;

(*iv*) — *хотя бы для одного номера k справедливо*

$$\alpha_k \triangleq \frac{(\nabla f_0(x^k),\ h^k)}{\|\nabla f_0(x^k)\| \|h^k\|} > 0,$$

то имеет место оценка

$$f_0(x^m) - f_0(x^*) \leqslant \left[c_1 \sum_{k=0}^{m-1} (\alpha_k)^2\right]^{-1},$$

где $f_0(x^*) = \min_{x \in R^n} f_0(x)$, $c_1 > 0$ — *константа.*

Если, кроме того, $f_0(x)$ — *сильновыпуклая функция, т. е. существует* $\beta > 0$ *такое, что*

$$f_0(\lambda x + (1 - \lambda) y) \leqslant \lambda f_0(x) + (1 - \lambda) f_0(y) - \lambda(1 - \lambda)\beta \|x - y\|^2,$$

$$\forall\, \lambda \in [0,\ 1],\ x,\ y \in R^n,$$

то имеют место оценки

$$f_0(x^m) - f_0(x^*) \leqslant (f_0(x^0) - f_0(x^*)) \exp\left[-\frac{\beta}{2\gamma}\sum_{k=0}^{m-1}(\alpha_k)^2\right];$$

$$\|x^m - x^*\|^2 \leqslant \frac{2}{\beta}(f_0(x^0) - f_0(x^*)) \exp\left[-\frac{\beta}{2\gamma}\sum_{k=0}^{m-1}(\alpha_k)^2\right].$$

Библиографические указания. Параграф основан на работе [192].

2.13. Методы случайного поиска

1. Алгоритм случайного поиска в выпуклых задачах минимизации

Задача 1. Найти $\arg\min_{x\in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположения 1. (*i*) — функция f_0 выпукла в R^n; (*ii*) — функция f_0 непрерывно дифференцируема в R^n и ее градиент удовлетворяет условию Липшица в R^n, т. е.

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma\|x - y\|,\ \gamma < \infty,\ \forall x,\ y \in R^n.$$

В методе случайного поиска на k-й итерации по известному приближению x^k вычисляется следующее приближение x^{k+1} как точка, в некотором смысле близкая к точке минимума функции f_0 на прямой $x^k - \rho h^k$, $\rho \in (-\infty, \infty)$, где h^k независимая реализация единичного случайного вектора равномерно распределенного на единичной сфере с центром в начале координат. Для выпуклой функции алгоритм генерирует последовательность $\{x^k\}_{k=0}^{\infty}$, для которой

$$f_0(x^k) - f_0(x^*) = O(1/k),$$

где x^*— решение задачи 1, $O(t)$ — величина порядка t, а для сильновыпуклой функции f_0 справедливо

$$x^k \to x^* \quad \text{и} \quad f_0(x^k) \to f_0(x^*) \quad \text{при} \quad k \to \infty$$

со скоростью сходимости геометрической прогрессии.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Положить $k = 0$.

Основной цикл. III. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Найти независимую реализацию h^k единичного случайного вектора ξ, равномерно распределенного на единичной n-мерной сфере с центром в начале координат.

V. Вычислить шаговый множитель ρ_k, удовлетворяющий неравенству

$$f_0(x^k - \rho_k h^k) \leqslant (1 - \lambda_k) f_0(x^k) + \lambda_k w_k, \qquad (2.20)$$

где λ_k — произвольная точка фиксированного отрезка $\lambda \leqslant \lambda_k \leqslant 1$ (λ — произвольная константа из полуинтервала $(0, 1]$), $w_k = \min\limits_{\rho \in (-\infty, \infty)} f_0(x^k - \rho h^k)$.

VI. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняется предположение 1 и пусть множество $X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0),\ x \in R^n\}$ ограничено. Тогда, если последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, такова, что $\nabla f_0(x^k) \neq 0$, $k = 0, 1, \ldots$, и что существует номер m_0, для которого $\alpha_{m_0} \neq 0$,*
где

$$\alpha_k \triangleq (\nabla f_0(x^k),\ h^k)/\|\nabla f_0(x^k)\|,$$

то для всех $m \geqslant m_0 + 1$ выполняется неравенство

$$f_0(x^m) - f_0(x^*) \leqslant c_1 \left[\sum_{i=0}^{m-1} (\alpha_k)^2\right]^{-1},$$

где c_1 — положительная постоянная.

Теорема 1'. *Пусть выполняются все предположения теоремы 1 и функция $f_0(x)$ сильновыпукла. Тогда для последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, выполняются неравенства*

$$f_0(x^m) - f_0(x^*) \leqslant (f_0(x^0) - f_0(x^*)) \exp\left(-\frac{\nu\lambda}{2\gamma} \sum_{i=0}^{m-1} (\alpha_i)^2\right),$$

$$m = 1, 2, \ldots;$$

$$\|x^m - x^*\|^2 \leqslant \frac{2}{\nu} (f_0(x^0) - f_0(x^*)) \exp\left(-\frac{\nu\lambda}{2\gamma} \sum_{i=0}^{m-1} (\alpha_i)^2\right),$$

$$m = 1, 2, \ldots,$$

здесь ν — параметр сильновыпуклой функции.

Замечание 1. При выполнении предположений теоремы 1 алгоритм 1 порождает последовательность $\{x^k\}_{k=0}^{\infty}$, для которой с вероятностью, равной единице, осуществляется событие «для всякого ε, удовлетворяющего неравенствам $0 < \varepsilon < 1/n$, найдется такое число $l(\varepsilon)$, что неравенства

$$f_0(x^k) - f_0(x^*) \leqslant 2\gamma\eta^2 n/(k\lambda(1 - n\varepsilon))$$

справедливы для всех целых $k \geqslant l(\varepsilon)$».

Отсюда, в частности, следует, что с вероятностью равной единице для любого числа $\beta > 2\gamma\eta^2 n/\lambda$ найдется число $\bar{l}(\beta)$ такое, что неравенства

$$f_0(x^k) - f_0(x^*) \leqslant \beta/k$$

справедливы для всех целых $k \geqslant \bar{l}(\beta)$.

Замечание 1'. При выполнении предположений теоремы 1' алгоритм 1 порождает последовательность $\{x^k\}_{k=0}^{\infty}$, для которой с вероятностью, равной единице, $x^k \to x^*$ и $f_0(x^k) \to f_0(x^*)$ при $k \to \infty$ со скоростью сходимости геометрической прогрессии.

2. Адаптивный алгоритм случайного поиска

Задача 2. Найти $\arg \min\limits_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

В данном адаптивном алгоритме случайного поиска величина шагового множителя изменяется в зависимости от результата, полученного на предыдущей итерации. Вектор ξ^k, определяющий направление движения в k-й итерации к следующему приближению x^{k+1}, является независимой реализацией случайного вектора ξ, распределенного по единичной сфере с плотностью $p_\xi(y)$

$$\int\limits_{\substack{\|y\|=1\\(g,y)>0}} p_\xi(y)\,ds = {}^1/_2, \quad \forall g \neq 0. \tag{2.21}$$

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, произвольное начальное значение шагового множителя ρ_0 и константы $\gamma_1 > 1$, $0 < \gamma_2 < 1$; положить $k = 0$.

Основной цикл. II. Найти независимую реализацию ξ^k случайного вектора ξ, распределенного по единичной сфере с плотностью $p_\xi(y)$, удовлетворяющей условию (2.21) (в частности [349], этому требованию удовлетворяют случайный вектор ξ, равномерно распределенный на сфере, а также проекция на единичную сферу вектора, равномерно распределенного в кубе или на его поверхности).

III. Если $f_0(x^k + \rho_k\xi^k) < f_0(x^k)$, то положить $h^k = \rho_k\xi^k$; иначе положить $h^k = 0$.

IV. Вычислить следующее приближение

$$x^{k+1} = x^k + h^k.$$

V. Вычислить следующее значение шагового множителя согласно условий

$$\rho_{k+1} = \begin{cases} \gamma_1\rho_k, & \text{если } f_0(x^{k+1}) < f_0(x^k), \\ \gamma_2\rho_k, & \text{если } f_0(x^{k+1}) \geqslant f_0(x^k). \end{cases}$$

VI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Если выполнены условия:* (**i**) — *функция* f_0 *ограничена снизу;* (**ii**) — *функция* f_0 *непрерывно дифференцируема;* (**iii**) — x^* — *единственная точка минимума функции* f_0; (**iv**) — $\nabla f_0(x) \neq 0$ *при* $x \neq x^*$; (**v**) — *множество* $\{x \mid f_0(x) \leqslant \alpha\}$ *ограничено для всех* $\alpha \leqslant f_0(x^0)$; (**vi**) — $\gamma_1\gamma_2 > 1$, *то последовательность* $\{x^k\}_{k=0}^{\infty}$,

порожденная алгоритмом 2, сходится с вероятностью 1 к точке минимума x^*.

Библиографические указания. При написании параграфа использовались работы [193, 349, 547, 6, 418, 57, 328, 224, 283, 284].

Глава 3

МЕТОДЫ ОПТИМИЗАЦИИ НЕДИФФЕРЕНЦИРУЕМЫХ ФУНКЦИЙ И МЕТОДЫ ОТЫСКАНИЯ СЕДЛОВЫХ ТОЧЕК

3.1. Методы обобщенного градиентного спуска

Задача 0. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной выпуклой функции $f_0 : R^n \to R^1$.

Предположение 0. Множество решений X^* непусто.

1. Алгоритм с постоянным шаговым множителем

В алгоритме 1 в k-й итерации за вектор движения h^k к следующему приближению x^{k+1} выбирается единичный вектор обобщенного градиента функции f_0 в точке x^k. Шаговый множитель ρ является постоянной величиной. При заданном $\delta > 0$ можно указать такое $\rho = \rho(\delta)$, что порожденная алгоритмом 1 последовательность $\{x^k\}_{k=0}^{\infty}$ попадает в область, где минимизируемая функция f_0 отличается от своего минимума на величину δ.

Алгоритм 1

Начало. I. Выбрать: произвольное начальное приближение $x^0 \in R^n$, постоянный шаговый множитель $\rho > 0$; положить $k = 0$.

Основной цикл. II. Вычислить обобщенный градиент $g(x^k)$ функции f_0 в точке x^k. Если $g(x^k) = 0$, то $x^k \in X^*$; иначе перейти к шагу III.

III. Вычислить вектор h^k (определяющий направление движения к следующему приближению x^{k+1})

$$h^k = g(x^k)/\|g(x^k)\|.$$

IV. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho h^k.$$

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Если f_0 выпуклая функция, то для произвольного $\delta > 0$ можно найти такое $\rho(\delta)$, что для последовательности x^0, x^1, ..., x^k, ..., порожденной алгоритмом 1, при $\rho = \rho(\delta)$ найдется такое $k = k^*$, что $x^{k^*} \in X^*$, либо такая подпоследовательность x^{k_0}, x^{k_1}, ..., x^{k_l}, ..., что*

$$\lim_{i \to \infty} f_0(x^{k_i}) - \min_{x \in R^n} f_0(x) < \delta.$$

Теорема 1'. *Если выпуклая функция f_0 имеет область минимумов X^*, содержащую сферу радиуса $r > \rho/2$, то для последовательности $x^0, x^1, \ldots, x^k, \ldots$, порожденной алгоритмом 1, найдется такое $k = k^*$, что $x^{k^*} \in X^*$.*

2. Основной алгоритм

В алгоритме 2 в k-й итерации за вектор, определяющий направление движения к следующему приближению x^{k+1}, выбирается единичный вектор обобщенного градиента функции f_0 в точке x^k. Шаговый множитель ρ_k удовлетворяет классическим условиям

$$\rho_k \geqslant 0; \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \lim_{k \to \infty} \rho_k = 0. \tag{3.1}$$

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$ и положить $k = 0$.

Основной цикл. II. Вычислить обобщенный градиент $g(x^k)$ функции f_0 в точке x^k. Если $g(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу III.

III. Вычислить вектор

$$h^k = g(x^k)/\|g(x^k)\|.$$

IV. Вычислить значение шагового множителя ρ_k, удовлетворяющее условиям теоремы 2.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

VI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Пусть $f_0(x)$ — выпуклая функция, область минимумов X^* которой ограничена. Тогда, если шаговые множители ρ_k, $k = 0, 1, \ldots$, таковы, что*

$$\rho_k \geqslant 0; \quad \lim_{k \to \infty} \rho_k = 0; \quad \sum_{k=0}^{\infty} \rho_k = \infty,$$

то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, удовлетворяет предельным соотношениям

$$\lim_{k \to \infty} \min_{x \in X^*} \|x^k - x\| = 0;$$

$$\lim_{k \to \infty} f_0(x^k) = \min_{x \in R^n} f_0(x).$$

Теорема 2'. *Пусть множество минимумов*

$$X^* \triangleq \{x \mid f_0(x) = \inf_{x \in R^n} f_0(x)\}$$

выпуклой функции f_0 непусто и шаговые множители ρ_k, $k = 0, 1, \ldots$, удовлетворяют условиям

$$\rho_k > 0,\ k = 0,\ 1,\ \ldots;\quad \lim_{k\to\infty} \rho_k = 0;\quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

*Тогда, если выполнено одно из следующих пяти условий: (**iv**) — множество X^* ограничено;*
(*v*) —

$$\sum_{k=0}^{\infty} \rho_k^2 < \infty;\quad (\boldsymbol{vi}) - \operatorname{int} X^* \neq \varnothing;$$

(***vii***) — *множество X^* является линейным многообразием в R^n;* (***viii***) — *$n = 2$, то предельные точки бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, принадлежат множеству X^*. При этом условия* (*v*) *и* (***vii***) *обеспечивают сходимость последовательности $\{x^k\}_{k=0}^{\infty}$ к точке $\bar{x} \in X^*$, а условие* (***vi***) *обеспечивает конечность алгоритма 2.*

3. Модификация основного алгоритма

В алгоритме 3 в k-й итерации за вектор направления движения к следующему приближению x^{k+1} выбирается вектор, обратный к обобщенному градиенту функции f_0 в точке x^k. Шаговый множитель ρ_k удовлетворяет классическим условиям (3.1).

Алгоритм 3

Начало. I. Выбрать произвольную начальную точку $x^0 \in R^n$ и положить $k = 0$.

Основной цикл. II. Вычислить обобщенный градиент $g(x^k)$ функции f_0 в точке x^k.

III. Найти значение шагового множителя ρ_k, удовлетворяющее условиям теоремы 3.

IV. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k g(x^k).$$

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 3. *Пусть $f_0(x)$ — выпуклая функция, область минимумов которой X^* ограничена. Тогда, если* (***i***) — *шаговые множители ρ_k, $k = 0, 1, \ldots$, таковы, что*

$$\rho_k \geqslant 0;\quad \lim_{k\to\infty} \rho_k = 0;\quad \sum_{k=0}^{\infty} \rho_k = \infty;$$

(***ii***) — *последовательность обобщенных градиентов $\{g(x^k)\}_{k=0}^{\infty}$, порожденная алгоритмом 3, ограничена, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$ удовлетворяет предельным соотношениям*

$$\lim_{k\to\infty} \min_{x\in X^*} \| x^k - x \| = 0;$$

$$\lim_{k\to\infty} f_0(x^k) = \min_{x\in R^n} f_0(x).$$

4. Первый алгоритм со специальным выбором шага

Алгоритм 4 может быть применен для минимизации функции $f_0(x)$, удовлетворяющей предположению 4.

Предположение 4. Пусть $f_0(x)$ — выпуклая функция такая, что при некотором $0 \leqslant \varphi < \frac{\pi}{2}$, для всех $x \in R^n$ выполняется неравенство

$$(g(x),\ x - x^*(x)) \geqslant \cos\varphi \|g(x)\| \|x - x^*(x)\|, \tag{3.2}$$

где $x^*(x)$ — точка, принадлежащая множеству минимумов функции $f_0(x)$ и лежащая на кратчайшем расстоянии от x.

В алгоритме 4 в k-й итерации за вектор направления движения h^k к следующему приближению x^{k+1} выбирается единичный вектор обобщенного градиента функции f_0 в точке x^k. Шаговые множители ρ_k, $k = 0, 1, \ldots$, вычисляются в соответствии с рекуррентной формулой

$$\rho_{k+1} = \rho_k \beta(\varphi), \quad \rho_0 = \rho_0(x^0,\ \varphi),$$

где

$$\beta(\varphi) = \begin{cases} \sin\varphi, & \frac{\pi}{4} \leqslant \varphi < \frac{\pi}{2}, \\ \frac{1}{2\cos\varphi}, & 0 \leqslant \varphi < \frac{\pi}{4}. \end{cases} \tag{3.3}$$

Последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 4, сходится к множеству минимумов со скоростью геометрической прогрессии и знаменателем $q = \beta(\varphi)$.

Алгоритм 4

Начало. I. Выбрать произвольную начальную точку $x^0 \in R^n$.

II. Найти угол $0 \leqslant \varphi < \frac{\pi}{2}$, удовлетворяющий неравенству (3.2) для всех $x \in R^n$.

III. Вычислить значение $\beta(\varphi)$ по формуле (3.3).

IV. Вычислить начальное значение шагового множителя ρ_0, удовлетворяющее неравенству

$$\rho_0 \geqslant \begin{cases} \|x^*(x^0) - x^0\| \cos\varphi, & \frac{\pi}{4} \leqslant \varphi < \frac{\pi}{2}, \\ \frac{\|x^*(x^0) - x^0\|}{2\cos\varphi}, & 0 \leqslant \varphi < \frac{\pi}{4}, \end{cases}$$

где $x^*(x^0)$ — точка, принадлежащая множеству минимумов функции $f_0(x)$ и лежащая на кратчайшем расстоянии от точки x^0.

V. Положить $k = 0$.

Основной цикл. VI. Вычислить вектор направления движения h^k к следующему приближению x^{k+1}

$$h^k = g(x^k) / \|g(x^k)\|.$$

VII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

VIII. Вычислить значение шагового множителя

$$\rho_{k+1} = \rho_k \beta(\varphi),$$

где $\beta(\varphi)$ определяется по формуле (3.3).

IX. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 4. *Пусть имеет место предположение 4. Тогда последовательность x^k, $k = 0, 1, \ldots$, порожденная алгоритмом 4, такова, что либо при некотором $\bar{k}$ — $g(x^{\bar{k}}) = 0$ и $x^{\bar{k}}$ принадлежит области минимумов, либо при всех $k = 0, 1, \ldots$ будет выполняться неравенство*

$$\| x^k - x^*(x^k) \| \leqslant \begin{cases} \rho_k / \cos\varphi, & \frac{\pi}{4} \leqslant \varphi < \frac{\pi}{2}, \\ 2\cos\varphi \cdot \rho_k, & 0 \leqslant \varphi < \frac{\pi}{4}. \end{cases}$$

Замечание 4. В неравенстве (3.2) $\cos\varphi$ показывает степень вытянутости поверхностей уровня функции f_0. Если в некоторой окрестности минимума функции f_0 не существует угла $\varphi < \pi/2$, удовлетворяющего неравенству (3.2), то такая функция называется существенно овражной и для ее минимизации алгоритм 4 неприменим. В этом случае следует использовать универсальный метод выбора шаговых множителей, как в алгоритме 3.

5. Второй алгоритм со специальным выбором шага

Алгоритм 5 может быть применен для минимизации функции f_0, удовлетворяющей предположениям 5.

Предположения 5. (*i*) — функция f_0 выпукла в R^n; (*ii*) — функция f_0 имеет единственную точку минимума x^*; (*iii*) — для любого числа $\alpha > 0$ существует конечное число $\tau > 0$ такое, что для любой пары точек x, $z \in Y$,

$$Y \triangleq \{y \mid \| y - x^* \| \leqslant \alpha\},$$

такой, что $f_0(x) = f_0(z) \neq f_0(x^*)$, выполняется условие

$$(\| x - x^* \| / \| z - x^* \|) \leqslant \tau.$$

В алгоритме 5 в k-й итерации за вектор направления движения h^k к следующему приближению x^{k+1} выбирается единичный вектор обобщенного градиента функции f_0 в точке x^k. Шаговые множители ρ_k, $k = 0, 1, \ldots$, вычисляются по рекуррентным формулам

$$\rho_{k+1} = \rho_k \frac{\sqrt{\sigma^2 - 1}}{\sigma}; \quad \rho_0 = \rho_0(x^0, \sigma),$$

где $\sigma \geqslant \sqrt{2}$ — характеризует степень «вытянутости» поверхностей уровня функции f_0.

Последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 5, сходится к точке минимума функции f_0 со скоростью геометрической прогрессии и знаменателем $q = \sqrt{\sigma^2 - 1}/\sigma$.

Алгоритм 5

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Найти число $\sigma \geqslant \sqrt{2}$ и значение шагового множителя $\rho_0 > 0$, удовлетворяющие условиям теоремы 5.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить вектор направления движения h^k к следующему приближению x^{k+1}

$$h^k = g(x^k)/\| g(x^k) \|.$$

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

VI. Вычислить значение шагового множителя

$$\rho_{k+1} = \rho_k \sqrt{\sigma^2 - 1}/\sigma.$$

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 5. *Пусть имеют место предположения 5 и пусть числа σ и ρ_0 удовлетворяют условиям:* (*i*) — $\sigma \geqslant \sqrt{2}$; (*ii*) — $\rho_0 \geqslant (\| x^0 - x^* \|)/\sigma$; (*iii*) — *для любой пары точек x, z принадлежащих множеству $Y \triangleq \{x \mid \| x - x^* \| \leqslant \sigma\rho_0\}$, и таких, что $f_0(x) = f_0(z) \neq f_0(x^*)$ выполняется условие*

$$(\| x - x^* \|)/(\| z - x^* \|) \leqslant \sigma.$$

Тогда последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 5, сходится к точке x^* со скоростью геометрической прогрессии

$$\| x^k - x^* \| \leqslant \rho_k \sigma, \quad k = 0, 1, \ldots,$$

знаменатель которой $q = \sqrt{\sigma^2 - 1}/\sigma$, за исключением случая, когда для некоторого $k = \bar{k}$, $g(x^{\bar{k}}) = 0$, т. е.

$$x^{\bar{k}} = x^*.$$

6. Алгоритм, использующий априорное знание минимума функции

Алгоритм 6 основан на априорном знании значения f_0^* минимизируемой функции в точке минимума. В k-й итерации за вектор движения h^k к следующему приближению x^{k+1} выбирается обобщенный градиент функции f_0 в точке x^k. Шаговые множители ρ_k, $k = 0, 1, \ldots$, вычисляются по формуле

$$\rho_k = \gamma (f_0(x^k) - f_0^*)/\| g(x^k) \|^2,$$

где γ — константа, удовлетворяющая неравенствам $0 < \gamma < 2$. Последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 6, сходится

к точке минимума x^* со скоростью геометрической прогрессии со знаменателем $q = (1 - \gamma(2 - \gamma)\alpha^2/\lambda^2)^{1/2} < 1$, где λ и α — некоторые константы.

Алгоритм 6

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$ и константу γ, $0 < \gamma < 2$; положить $k = 0$.

Основной цикл. II. Вычислить вектор направления движения h^k к следующему приближению x^{k+1} по формуле $h^k = g(x^k)$.

III. Вычислить значение шагового множителя

$$\rho_k = \gamma(f_0(x^k) - f_0^*)/\|h^k\|^2.$$

IV. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 6. *Пусть выпуклая функция f_0 имеет единственную точку минимума x^*, причем известно значение f_0^* функции f_0 в точке x^*. Тогда, при $0 < \gamma < 2$ последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 6, сходится к x^* из любого начального приближения $x^0 \in R^n$.*

Теорема 6'. *Пусть: (i) — $f_0(x)$ — сильновыпуклый функционал, причем*

$$f_0(x) - f_0(x^*) \geq \alpha\|x - x^*\|^2, \quad \alpha > 0;$$

(ii) — функция $f_0(x)$ удовлетворяет условию Липшица на области

$$Y \triangleq \{x \mid (\|x - x^*\|) \leq (\|x^0 - x^*\|)\},$$

т. е. для всех $x, y \in Y$ выполняется

$$f_0(x) - f_0(y) \leq \lambda\|x - y\|.$$

Тогда, при $0 < \gamma < 2$ последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 6, сходится к точке минимума x^* со скоростью геометрической прогрессии

$$\|x^k - x^*\| \leq q^k\|x^0 - x^*\|$$

со знаменателем $q = (1 - \gamma(2 - \gamma)\alpha^2/\lambda^2)^{1/2} < 1$.

7. Помехоустойчивый алгоритм

Алгоритм 7 указывает на устойчивость метода обобщенного градиентного спуска к малым ошибкам в вычислении точек x^k, $k = 0, 1, \ldots$ и обобщенных градиентов $g(x^k)$ в этих точках. В k-итерации алгоритма за вектор движения h^k к следующему приближению x^{k+1} выбирается единичный вектор обобщенного градиента функции f_0 в точке y^k, лежащей в δ_k-окрестности точки x^k. Последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 7, сходится

к точке $x^* \in X^*$,

$$X^* = \{x \mid f_0(x) = f_0^*\};$$

$$f_0^* = \min_{x \in R^n} f_0(x).$$

Алгоритм 7

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$ и положить $k = 0$.

Основной цикл. II. Вычислить шаговый множитель ρ_k и величину смещения δ_k, удовлетворяющие условиям теоремы 7.

III. Вычислить обобщенный градиент $g(y^k)$ функции f_0 в любой точке y^k, удовлетворяющей неравенству

$$\| y^k - x^k \| \leqslant \delta_k.$$

IV. Вычислить вектор направления движения h^k к следующему приближению x^{k+1}

$$h^k = g(y^k) / \| g(y^k) \|.$$

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

VI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 7. *Пусть f_0 — выпуклая функция, множество минимумов X^* которой непусто. Тогда, если числа δ_k, ρ_k, $k = 0, 1, \ldots,$ выбирать такими, что*

$$\rho_k > 0; \quad \delta_k > 0; \quad \lim_{k \to \infty} \delta_k = 0;$$

$$\sum_{k=0}^{\infty} \delta_k \rho_k < \infty; \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 7, удовлетворяет предельному соотношению

$$\lim_{k \to \infty} x^k = x^* \in X^*.$$

8. Многошаговый метод обобщенного градиентного спуска

Предположение 8. Функция f_0 — выпуклая.

В приводимом ниже алгоритме направление спуска выбирается с использованием обобщенных градиентов и значений функции f_0 на предыдущих итерациях. На каждой итерации требуется решать специальную задачу минимизации, которая соответствующей нормировкой сводится к задаче линейного программирования. Шаговые множители ρ_k удовлетворяют классическим условиям.

Алгоритм 8

Начало. I. Выбрать произвольное натуральное число $m \geqslant 1$.

II. Выбрать произвольный набор точек $\{x^{-m+1}, \ldots, x^0\}$.

III. Выбрать константу α, $\alpha > f_0(x^0)$.

IV. Положить $k = 0$.

Основной цикл. V. Вычислить $\hat{\nabla} f_0(x^k)$ — обобщенный градиент функции f_0 в точке x^k.

VI. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям теоремы 8.

VII. Если $\hat{\nabla} f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VIII.

VIII. Если $f_0(x^k) > \alpha$, то положить $\mathcal{J}_k = \{k\}$ и перейти к шагу IX, если $f_0(x^k) \leqslant \alpha$, то положить $\mathcal{J}_k = \{k - m + 1, \ldots, k\}$ и перейти к шагу IX.

IX. Вычислить вектор h^k из условия

$$\min_{\|h\| \leqslant \rho_k} \varphi_k(h) = \varphi_k(h^k),$$

где функция

$$\varphi_k(h) \triangleq \max_{j \in \mathcal{J}_k} [f_0(x^j) + (x^k - x^j, \hat{\nabla} f_0(x^j)) + (h, \hat{\nabla} f_0(x^j))].$$

X. Вычислить следующее приближение

$$x^{k+1} = x^k + h^k.$$

XI. Положить $k = k + 1$ и перейти к шагу V.

Теорема 8. *Пусть выполняются условия:* (*i*) — *функция* f_0 *выпукла;* (*ii*) — *множество*

$$X^* \triangleq \{\bar{x} \mid f_0(\bar{x}) = \inf_{x \in R^n} f_0(x)\}$$

непусто и ограничено; (*iii*) — *шаговые множители* ρ_k, $k = 0, 1, \ldots$, *удовлетворяют условиям*

$$\rho_k \to +0 \text{ при } k \to \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

Тогда бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 8, такова, что*

$$\min_{x \in X^*} \| x - x^k \| \to 0 \text{ при } k \to \infty;$$

$$f_0(x^k) \to \inf_{x \in R^n} f_0(x) \text{ при } k \to \infty.$$

Замечание 8. Если множество X^* содержит внутренние точки, то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 8, конечна.

9. ε-субградиентный метод

Задача 9. Найти $\arg \min_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположения 9. (*i*) — функция f_0 — выпуклая полунепрерывная снизу;
(*ii*) — $\inf_{x \in R^n} f_0(x) > -\infty$; (*iii*) — по крайней мере в одной точке $\bar{x} \in R^n$ выполняется условие $f_0(\bar{x}) < \infty$.

Определение 9. Для произвольного $\varepsilon > 0$ вектор $g_\varepsilon(x) \in R^n$ называется ε-субградиентом функции f_0 в точке x, если $f_0(z) \geqslant f_0(x) - \varepsilon + (z - x, g_\varepsilon(x))$ при всех $z \in R^n$. Множество ε-субградиентов в точке x обозначается через $G_\varepsilon(x)$.

Алгоритм 9

Начало. I. Выбрать вектор $x^0 \in R^n$ такой, что $f_0(x^0) < \infty$.

II. Выбрать константы $\varepsilon_0 > 0$ и $0 < \alpha < 1$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить

$$\varepsilon_{k+1} = \alpha^i \varepsilon_k,$$

где i — наименьшее неотрицательное целое число, при котором $0 \notin G_{\varepsilon_{k+1}}(x^k)$.

(Если x^k не является точкой минимума функции f_0, то всегда существует неотрицательное целое число i, при котором выполняется включение $0 \notin G_{\alpha^i \varepsilon_k}(x^k)$).

V. Найти вектор h^k, для которого выполняется неравенство

$$\sup_{g \in G_{\varepsilon_{k+1}}(x^k)} (h^k, g) < 0.$$

VI. Положить $x^{k+1} = x^k + \rho_k h^k$, где $\rho_k > 0$ такое, что выполняется неравенство $f_0(x^k) - f_0(x^{k+1}) > \varepsilon_{k+1}$.

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 9. *Пусть выполняются предположения 9. Тогда либо бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$ удовлетворяет предельному соотношению*

(*iv*) —

$$\lim_{k \to \infty} f_0(x^k) = \min_{x \in R^n} f_0(x),$$

либо $f_0(x^m) = \min_{x \in R^n} f_0(x)$ при некотором $m > 0$. Если, кроме того, множество

$$X^* \triangleq \{x^* \mid f_0(x^*) = \min_{x \in R^n} f_0(x)\}$$

не пусто и ограничено, то: (*v*) — *каждая сходящаяся подпоследовательность последовательности $\{x^k\}_{k=0}^{\infty}$ имеет предел в X^* и хотя бы одна такая подпоследовательность существует;* (*vi*) — *при каждом $\varepsilon > 0$ существует $\bar{m} \geqslant 0$ такое, что $x^k \in X^* + \varepsilon Y$ при всех*

$k \geqslant \overline{m}$, где $Y \triangleq \{x \mid \|x\| \leqslant 1\}$; (vii) — если минимум функции f_0 достигается в единственной точке x^*, то $\{x^k\}_{k=0}^{\infty}$ стремится к x^*.

Библиографические указания. При написании параграфа использовались работы [395, 396, 397, 401, 404, 391, 392, 406, 290, 126, 160, 277, 433, 491, 481].

3.2. Методы градиентного типа с растяжением пространства

З а д а ч а 0. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной почти дифференцируемой функции $f_0 : R^n \to R^1$.

Сущность методов градиентного типа с растяжением пространства заключается в построении в процессе последовательных приближений линейных операторов, изменяющих метрику пространства, и выборе направления спуска, соответствующего антиградиенту в пространстве с новой метрикой.

Определение 0. Оператором растяжения пространства R^n в направлении $\xi \in R^n$ ($\|\xi\| = 1$) с коэффициентом α ($\alpha \geqslant 0$), называется оператор $G_\alpha(\xi)$, действующий на вектор x, представленный в форме

$$x = \gamma_\xi(x)\xi + d_\xi(x);$$
$$(\xi, d_\xi(x)) = 0,$$

(здесь $\gamma_\xi(x) = (x, \xi)$; $d_\xi(x) = x - (x, \xi)\xi$) следующим образом:

$$G_\alpha(\xi)x = \alpha\gamma_\xi(x)\xi + d_\xi(x) = x + (\alpha - 1)(x, \xi)\xi,$$

где $G_\alpha(\xi)$ — линейный симметричный оператор.

Оператор $G_\beta(\xi) = G_{1/\alpha}(\xi)$ называется оператором «сжатия».

Ниже приводятся алгоритмы с растяжением пространства в направлениях: сначала почти градиента, а затем разности двух последовательных почти градиентов (§ 3.3).

В алгоритме 1 в k-й итерации следующее приближение x^{k+1} находят по формуле

$$x^{k+1} = x^k - B_k\rho_k\xi^k,$$

где ξ^k — единичный вектор почти градиента функции $\varphi_k(y) \triangleq \triangleq f_0(B_k y)$, которая получается из $f_0(x)$ при использовании линейного преобразования пространства $y = A_k x$; B_k — оператор, обратный результирующему оператору A_k преобразования пространства (A_k получается в результате последовательного применения операторов растяжения пространства в направлении нормированных почти градиентов $\xi^0, \xi^1, \ldots, \xi^{k-1}$ с коэффициентами $\alpha_1, \ldots, \alpha_k$: $A_k = G_{\alpha_k}(\xi^{k-1})A_{k-1}$); ρ_k — шаговый множитель.

Операторы B_{k+1} для отображения преобразованного в основное пространство R^n определяются рекуррентными соотношениями

$$B_{k+1} = B_k G_{\beta_{k+1}} \xi^k); \quad B_0 = I,$$

где $\beta_{k+1} \triangleq 1/\alpha_{k+1}$ — коэффициенты «сжатия» пространства.

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in R^n$ и неособую матрицу B_0 (можно выбрать $B_0 = I$, где I — единичная $n \times n$-матрица); положить $k = 0$.

Основной цикл. II. Вычислить почти градиент $g(x^k)$ функции f_0 в точке x^k.

III. Если $g(x^k) = 0$, то положить $x^* = x^k$ и закончить вычисления; иначе перейти к шагу IV.

IV. Вычислить оператор B_k^T, сопряженный оператору B_k.

V. Вычислить почти градиент $\tilde{g}(y^k)$ функции $\varphi_k(y) \triangleq f_0(B_k y)$ в точке $y^k = B_k^{-1} x^k$

$$\tilde{g}(y^k) = B_k^T g(x^k).$$

VI. Вычислить направление растяжения пространства

$$\xi^k = \tilde{g}(y^k)/\|\tilde{g}(y^k)\|.$$

VII. Вычислить значения шагового множителя ρ_k.

VIII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k B_k \xi^k.$$

IX. Найти коэффициент растяжения пространства α_{k+1}.

X. Вычислить коэффициент «сжатия» пространства

$$\beta_{k+1} = 1/\alpha_{k+1}.$$

XI. Вычислить оператор B_{k+1}, обратный результирующему оператору A_{k+1} преобразования пространства,

$$B_{k+1} = B_k G_{\beta_{k+1}}(\xi^k),$$

где оператор «сжатия» $G_{\beta_{k+1}}(\xi^k)$ вычисляется согласно определения 0.

XII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Пусть $f_0(x)$ — почти дифференцируемая функция и x^* — точка ее локального минимума. Тогда, если в процессе реализации алгоритма 1 выполняются условия:*

(i) —

$$\|x^k - x^*\| \leqslant \sigma, \quad \sigma > 0, \quad k = 1, 2, \ldots;$$

(ii) —

$$1 + \delta \leqslant \alpha_k \leqslant \bar{\alpha}, \quad \delta > 0,$$

то существует такая подпоследовательность $\{x^{k_s}\}_{s=1}^{\infty}$ и число $\gamma > 0$, что

$$\|\tilde{g}(y^{k_s})\| \leqslant \gamma \left(\prod_{j=1}^{k_s} \alpha_j \right)^{-\frac{1}{n}}, \quad s = 1, 2, \ldots .$$

Теорема 1'. *Пусть $f_0(x)$ — почти дифференцируемая функция, определенная в некоторой сферической окрестности S_σ точки x^**

локального минимума, и в тех точках, где функция дифференцируема, ее производная $f'_{0,\mu(x)}(x)$ *по направлению* $\mu(x) = x - x^*$ *удовлетворяет неравенству*

$$\lambda \leqslant \frac{f'_{0,\mu(x)}(x)}{f_0(x) - f_0(x^*)} \leqslant \lambda', \quad \lambda' > \lambda > 0, \quad x \neq x^*.$$

Тогда, если в алгоритме 1 принять

(*i*) —
$$x^0 \in S_\sigma;$$

(*ii*) —
$$\rho_k = \frac{2\lambda\lambda'}{\lambda + \lambda'} \frac{f_0(x^k) - f_0(x^*)}{\|\tilde{g}(y^k)\|};$$
$$\alpha_{k+1} \equiv \alpha = \frac{\lambda' + \lambda}{\lambda' - \lambda},$$

то найдутся константа γ' *и подпоследовательность индексов* $\{k_s\}_{s=1}^{\infty}$, $k_s < k_{s+1}$, *такие, что*

$$f_0(x^{k_s}) - f_0(x^*) \leqslant \gamma' \alpha^{-k_s/n}, \quad s = 1, 2, \ldots .$$

Замечание 1. Описанный в теореме 1′ способ выбора коэффициента растяжения α_k, $k = 1, 2, \ldots$, и шагового множителя ρ_k, $k = 0, 1, 2, \ldots$, находит непосредственное применение при решении систем нелинейных уравнений

$$f_i(x) = 0, \quad i = 1, \ldots, n,$$

путем сведения к задаче $\min\limits_x \max\limits_{1 \leqslant i \leqslant n} |f_i(x)|$.

При этом, как показано в [148], в регулярном случае при достаточно хорошем начальном приближении x^0 константы λ и λ' можно выбирать близкими к единице, что обеспечивает быструю сходимость.

При решении выпуклых задач, в общем случае $f_0(x^*)$ неизвестно, поэтому представляет интерес вопрос о подборе $f_0(x^*)$ в процессе счета. Справедлива следующая теорема.

Теорема 1″. *Пусть выпуклая функция* $f_0(x)$ *обладает следующими свойствами:* (*i*) — *существует постоянная* $\bar{\gamma} > 1$ *такая, что если функция*

$$\varphi(\tau), \quad \varphi(\tau) \triangleq f_0((1-\tau)x^1 + \tau x^2), \quad 0 \leqslant \tau \leqslant 1,$$

строго убывает по τ, *то выполняется неравенство*

$$f'_{0,(x^1-x^2)}(x^1) \leqslant \bar{\gamma}(f_0(x^1) - f_0(x^2)),$$

где $f'_{0,(x^1-x^2)}(x^1)$ — *производная функции* $f_0(x)$ *по направлению* $(x^1 - x^2)$ *в точке* x^1;

(*ii*) —
$$\lim_{\|x\| \to \infty} f_0(x) = +\infty.$$

Тогда, если в алгоритме 1 принять

$$\alpha_{k+1} = (\bar{\gamma} + 1)/(\bar{\gamma} - 1);$$

$$\rho_k = 2\bar{\gamma}(f_0(x^k) - \bar{f})/(\bar{\gamma}+1)\|\tilde{g}(y^k)\|,$$

где $\bar{f}$ выбрано большим или равным $f^* \triangleq \min_{x \in R^n} f_0(x)$, то последовательность $\{\rho_k\}_{k=0}^{\infty}$ является ограниченной и для произвольного $\varepsilon > 0$ найдется $\bar{k}$ такое, что

$$f_0(x^{\bar{k}}) \leqslant \bar{f} + \varepsilon$$

(счет прекращается, если на некотором шаге $f_0(x^{\bar{k}}) \leqslant \bar{f}$). Если $\bar{f}$ выбрано меньшим f^*, то последовательность $\{\rho_k\}_{k=0}^{\infty}$ является неограниченной.

Замечание 1'. Теорема 1'' позволяет строить алгоритм минимизации функции $f_0(x)$, удовлетворяющей условиям теоремы, при неизвестном f^*. Этот алгоритм связан с подбором $\bar{f}^*$ с использованием следующих признаков: если при некотором $\bar{f}$ шаговый множитель ρ_k превосходит наперед выбранное достаточно большое число, то $\bar{f}$ увеличивается; при приближении $f_0(x^k)$ к $\bar{f}$ — значение $\bar{f}$ уменьшается.

Теорема 1'''. *Пусть функция f_0 выпукла в R^n и в процессе применения алгоритма 1 выполняются следующие условия:*

(*i*) — $\|g(x^k)\| \leqslant \sigma, \quad k = 0, 1, \ldots;$

(*ii*) — $1 < \alpha \equiv \alpha_k; \quad \beta_k = 1/\alpha_k.$

Тогда для последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, справедливы неравенства

$$\min_{1 \leqslant r \leqslant k} \|\tilde{g}(y^r)\| \leqslant \sigma\sqrt{k(\alpha^2-1)}/\sqrt{\alpha^{2k/n}-1}, \quad k = 1, 2, \ldots .$$

Библиографические указания. При написании параграфа использовались работы [398, 401, 403, 407, 400, 408, 409, 405, 340].

3.3. Методы градиентного типа с растяжением пространства в направлении разности двух последовательных почти градиентов ($r(\alpha)$-алгоритм)

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, коэффициент растяжения пространства $\alpha > 1$ и положить $\tilde{g}^0 = 0$ (0 — n-мерный нуль-вектор), $B_0 = I$ (I — единичная матрица), $k = 0$.

Основной цикл. II. Вычислить почти градиент $g(x^k)$ функции f_0 в точке x^k; если он определен неоднозначно, то взять такой почти градиент, что $(B_k\tilde{g}^k, g(x^k)) \leqslant 0$.

III. Вычислить вектор

$$\bar{g}^k = B_k^T g(x^k).$$

Примечание: вектор $\bar{g}^k$ является почти градиентом функции $\varphi_k(y) \triangleq f_0(B_k y)$ в точке $y = A_k x^k$, где $A_k \triangleq B_k^{-1}$ — оператор растяжения пространства после k-й итерации.

IV. Вычислить разность двух почти градиентов от функции $\varphi_k(y)$, вычисленных в точках $y^k = A_k x^k$, $\tilde{y}^k = A_k x^{k-1}$, по формуле

$$r^k = \bar{g}^k - \tilde{g}^k.$$

V. Вычислить направление растяжения пространства

$$\xi^k = r^k / \| r^k \|.$$

VI. Вычислить оператор B_{k+1}, обратный результирующему оператору A_{k+1} преобразования пространства после $k+1$-й итерации,

$$B_{k+1} = B_k G_\beta(\xi^k),$$

где $\beta = 1/\alpha$ — коэффициент «сжатия» пространства.

VII. Вычислить почти градиент $\tilde{g}^{k+1}$ функции $\varphi_{k+1}(y) \triangleq \triangleq f_0(B_{k+1} y)$ в точке $y = B_{k+1}^{-1} x^k$:

$$\tilde{g}^{k+1} = G_\beta(\xi^k)\, \bar{g}^k = B_{k+1}^T g(x^k).$$

VIII. Вычислить значение шагового множителя ρ_k из условия

$$\rho_k = \arg \overline{\min_{\rho \geqslant 0}} f_0(x^k - \rho B_{k+1} \tilde{g}^{k+1}) \qquad (3.4)$$

(операция $\overline{\min}$ означает определение ближайшего к нулю локального минимума).

IX. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k B_{k+1} \tilde{g}^{k+1}.$$

Примечание: шаг IX фактически реализует шаг наискорейшего спуска для функции $\varphi_{k+1}(y) \triangleq f_0(B_{k+1} y)$:

$$y^{k+1} = \tilde{y}^k - \rho_k \tilde{g}^{k+1}.$$

X. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Если выполнены условия: (i) — функция f_0 — непрерывна и кусочно-гладкая; (ii) — $\lim_{\|x\| \to \infty} f_0(x) = +\infty$; (iii) — функция f_0 обладает следующим свойством: в любой ограниченной области X для любого $\delta > 0$ найдется такое $\varepsilon > 0$, что если x, $y \in X$, $f_0(x) - f_0(y) < \varepsilon$ и на отрезке между x и y функция f_0 убывает, то $\| x - y \| < \delta$ (для выпуклых функций это свойство является аналогом сильной выпуклости), то предельной точкой последовательности*

$\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1 при любом* $\alpha > 1$, *является некоторая точка* $\bar{x}$, *множество почти градиентов которой образует линейно-зависимое семейство векторов. При этом, в силу монотонности процесса спуска* ($f_0(x^0) \geqslant f_0(x^1) \geqslant \ldots \geqslant f_0(x^k) \geqslant \ldots$) *последовательность* $\{f_0(x^k)\}_{k=0}^{\infty}$ *сходится к* $f_0(\bar{x})$.

Теорема 1'. *Если выполняются все условия теоремы 1 и* x^* — *изолированная точка локального минимума,* x^0 — *такая точка, что связная компонента множества* $\{x \mid f_0(x^*) \leqslant f_0(x) \leqslant f_0(x^0)\}$, *содержащая точки* x^*, x^0, *не имеет кроме* x^* *других точек* z, *у которых семейство почти градиентов* $\hat{G}(z)$ *линейно-зависимо, то последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1 с начальной точкой* x^0, *удовлетворяющей условиям теоремы, сходится к* x^*.

Замечание 2. В отличие от других градиентных методов минимизации негладких функций (обобщенного градиентного спуска, обобщенного градиентного спуска с растяжением пространства в направлении почти градиента) $r(\alpha)$-алгоритм обеспечивает монотонность спуска благодаря определенному по формуле (3.4) способу выбора шага спуска: $f_0(x^0) \geqslant f_0(x^1) \geqslant \ldots \geqslant f_0(x^k)$. Однако он существенно отличается от обычных релаксационных методов следующим: если $\rho_k = 0$, то это не значит, что процесс спуска прекращается. При выполнении серии итераций с нулевым шагом точка x^k стоит на месте, но изменяются B_k и $\tilde{g}^k$. В это время как бы в скрытой форме осуществляется поиск подходящего направления спуска.

Библиографические указания. При написании параграфа использовались работы [398, 401, 403, 407, 400, 408, 409, 405, 340].

3.4. Методы локального случайного поиска

З а д а ч а 0. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной непрерывной функции $f_0 : R^n \to R^1$.

1. Алгоритм локального случайного поиска с парной пробой

Алгоритм 1

Н а ч а л о. I. Выбрать: произвольное начальное приближение $x^0 \in R^n$, пробный шаговый множитель $\gamma > 0$ и рабочий шаговый множитель $\rho > 0$; положить $k = 0$.

О с н о в н о й ц и к л. II. Вычислить независимую реализацию ξ^k случайного единичного вектора ξ, равномерно распределенного по всем направлениям пространства параметров x.

III. Вычислить вектор движения h^k к следующему приближению x^{k+1}

$$h^k = \xi^k \operatorname{sign}[f_0(x^k - \gamma\xi^k) - f_0(x^k + \gamma\xi^k)].$$

IV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho h^k.$$

V. Положить $k = k + 1$ и перейти к шагу II.

Замечание 1. Характерной особенностью данного алгоритма является его повышенная тенденция к «блужданию», даже в том случае, если решение задачи 0 найдено. Однако алгоритм 1 приводит с большой вероятностью в ε-окрестность точек локального минимума, зависящую от шага ρ, размерности пространства R^n и вида функции f_0. (Например, в центральном поле, т. е. когда $f_0(x) = x - x^*$, для того чтобы попасть в ε-окрестность точки x^* с большой вероятностью, необходимо шаговый множитель ρ выбирать из условия $\rho \leqslant \varepsilon/\alpha_n^*$, где при $n = 2$ $\alpha_n^* = 0{,}7$; при $n = 3$ $\alpha_n^* = 0{,}9$; при $n = 4$ $\alpha_n^* = 1{,}0$).

2. Алгоритм локального случайного поиска с возвратом при неудачном шаге

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, шаговый множитель $\rho > 0$; положить $k = 0$.

Основной цикл. II. Вычислить независимую реализацию ξ^k случайного единичного вектора ξ, равномерно распределенного по всем направлениям пространства R^n.

III. Вычислить вектор z по формуле $z = x^k + \rho\xi^k$.

IV. Если $f_0(z) < f_0(x^k)$, то положить $x^{k+1} = z$ и перейти к шагу V; иначе положить $x^{k+1} = x^k$ и перейти к шагу V.

V. Положить $k = k + 1$ и перейти к шагу II.

Замечание 2. Алгоритм 2 (как и алгоритм 1) может быть рекомендован для оптимизации объектов, функция качества которых изменяется во времени со сравнительно большой скоростью.

3. Алгоритм локального случайного поиска с линейной экстраполяцией

Смысл такого алгоритма поиска сводится к следующему. После неудачного случайного шага делается двойной шаг в противоположном направлении, но функция цели f_0 в этом состоянии не вычисляется, а экстраполируется в предположении о линейном характере функции цели f_0.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, шаговый множитель $\rho > 0$; вычислить $f_0(x^0)$; положить $k = 0$.

Основной цикл. II. Вычислить независимую реализацию ξ^k случайного единичного вектора ξ, равномерно распределенного по всем направлениям пространства R^n.

III. Вычислить вектор $z = x^k + \rho\xi^k$.

IV. Вычислить $f_0(z)$ — значение функции f_0 в точке z, если $f_0(z) < f_0(x^k)$, то положить $x^{k+1} = z$, $f_0(x^{k+1}) = f_0(z)$ и перейти к шагу VII; иначе перейти к шагу V.

V. Вычислить вектор $x^{k+1} = x^k - \rho\xi^k$.

VI. Вычислить «значение» функции f_0 в точке x^{k+1}

$$f_0(x^{k+1}) = 2f_0(x^k) - f_0(z).$$

VII. Положить $k = k + 1$ и перейти к шагу II.

Замечание 3. Алгоритм 3 обладает повышенным быстродействием по сравнению с алгоритмами 1 и 2, ибо он использует неблагоприятные шаги. Однако он плохо работает в нелинейной задаче.

4. Алгоритм случайного поиска по наилучшей пробе с накоплением

Сущность алгоритма заключается в следующем. В k-й итерации из точки x^k делается m ($m \geqslant 1$) пробных шагов по независимым случайным направлениям $\xi^{k,j}$, $j = 1, \ldots, m$, и определяется наилучшее направление ξ^{k,j^*}. Рабочий шаг делается именно в этом направлении.

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, пробный шаговый множитель $\gamma > 0$ и рабочий шаговый множитель $\rho > 0$, число пробных шагов $m \geqslant 1$; положить $k = 0$.

Основной цикл. II. Вычислить m независимых реализаций $\xi^{k,i}$, $i = 1, \ldots, m$, случайного единичного вектора ξ, равномерно распределенного по всем направлениям пространства R^n.

III. Вычислить индекс i^*, удовлетворяющий условию

$$f_0(x^k + \gamma\xi^{k,i^*}) = \min_{i=1,\ldots,m} f_0(x^k + \gamma\xi^{k,i}).$$

IV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho\xi^{k,i^*}.$$

V. Положить $k = k + 1$ и перейти к шагу II.

Замечание 4. С увеличением числа пробных шагов направление вектора движения ξ^{k,i^*} приближается к направлению, обратному градиенту $\nabla f_0(x^k)$, и в пределе при $m \to \infty$, $\gamma \to 0$ совпадает с ним.

5. Алгоритм статистического градиента

Алгоритм 5

Начало. I. Выбрать произвольную начальную точку $x^0 \in R^n$, пробный шаговый множитель $\gamma > 0$, рабочий шаговый множитель $\rho > 0$, число пробных шагов $m \geqslant 1$; положить $k = 0$.

Основной цикл. II. Вычислить m независимых реализаций $\xi^{k,i}$, $i = 1, \ldots, m$, случайного единичного вектора ξ, равномерно распределенного по всем направлениям пространства R^n.

III. Вычислить приращения функции f_0

$$\Delta_i = f_0(x^k + \gamma\xi^{k,i}) - f_0(x^k), \quad i = 1, \ldots, m.$$

IV. Вычислить вектор $g^k = \sum_{i=1}^{m} (\xi^{k,i}\Delta_i)$.

V. Вычислить вектор движения h^k к следующему приближению x^{k+1} по формуле $h^k = -g^k / \| g^k \|$.

VI. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho h^k.$$

VII. Положить $k = k + 1$ и перейти к шагу II.

Замечание 5. Вектор g^k является статистической оценкой градиентного направления функции f_0 в точке x^k (при $m \to \infty$, $\gamma \to 0$ g^k стремится к направлению градиента $\nabla f_0 (x^k)$).

Библиографические указания. При написании параграфа использована работа [326]. Дополнительные сведения о методах локального случайного поиска можно найти в работах [326—328, 177, 178] и в сборнике статей [4].

3.5. Псевдоградиентные методы адаптации и обучения

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной функции $f^1 : R^n \to R^1$.

Псевдоградиентные алгоритмы адаптации и обучения основаны на предположении, что существует некоторая детерминированная гладкая функция $f_1 : R^n \to R^1$, называемая *критерием оптимальности.* Этот критерий либо задан априорно (если, например, исходная задача заключается в минимизации функции $f_0(x) \equiv f_1(x)$), либо может вводиться искусственно. Если минимизируемая функция f_0 — гладкая, то $f_1(x) \equiv f_0(x)$ и в псевдоградиентном алгоритме в качестве вектора ξ^k, определяющего направление движения к следующему приближению x^{k+1}, выбирается псевдоградиент функции f_0. Если минимизируемая функция f_0 — негладкая, то в качестве вектора ξ^k выбирается псевдоградиент некоторой специально построенной вспомогательной гладкой функции f_1, точки минимума которой являются точками минимума функции f_0.

Например, если точка x^* является точкой минимума функции f_0, то в качестве функции f_1 можно выбрать $f_1(x) = \| x - x^* \|$. Чтобы этот пример не показался бесполезным и не вызвал недоразумений (ведь нам неизвестна точка минимума функции f_0), следует отметить одно чрезвычайно важное свойство псевдоградиентных алгоритмов — эти алгоритмы не требуют вычисления ни значений $f_1(x)$, ни градиентов $\nabla f_1(x)$ для функции f_1, они требуют только вычисления псевдоградиентов ξ^k для функции f_1 в точках $x = x^k$, а псевдоградиент ξ^k часто можно вычислить косвенным путем, не зная точки x^*.

Определение 1. Вектор ξ^k называется псевдоградиентом функ-

ции f_1 в точке $x = x^k$, если он является реализацией некоторого случайного вектора ξ, удовлетворяющего условию

$$(\nabla f_1(x^k),\ E\xi) \geqslant 0.$$

Приводимые ниже теоремы о сходимости псевдоградиентного алгоритма адаптации и обучения позволяют с единой точки зрения обосновать сходимость многих алгоритмов стохастической аппроксимации, случайного поиска, обобщенного градиентного поиска и др.

Предположения 1. (*i*) — функция f_1 ограничена снизу

$$f_1(x) \geqslant f_1^* > -\infty;$$

(*ii*) — градиент функции f_1 удовлетворяет условию Липшица

$$\|\nabla f_1(x) - \nabla f_1(y)\| \leqslant \gamma \|x - y\|, \quad \forall x,\ y \in R^n, \quad \gamma < \infty.$$

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^1 \in R^n$.

II. Положить $k = 1$.

Основной цикл. III. Найти значения ρ_k и λ_k, удовлетворяющие условиям теоремы 1.

IV. Вычислить псевдоградиент ξ^k функции f_1 в точке x^k, т. е. вычислить одну реализацию ξ^k случайного вектора $\bar{\xi}^k$, условное математическое ожидание которого удовлетворяет неравенству

$$(\nabla f_1(x^k),\ E(\bar{\xi}^k/\xi^1,\ \xi^2,\ \ldots,\ \xi^{k-1})) \geqslant 0,$$

(Предполагается, что $\bar{\xi}^k$ и f_1 удовлетворяют неравенству

$$E(\|\bar{\xi}^k\|^2/\xi^1,\ \ldots,\ \xi^{k-1}) \leqslant \lambda_k + \delta_1 f_1(x^{k-1}) + \\ + \delta_2(\nabla f_1(x^{k-1}),\ E(\bar{\xi}^k/\xi^1,\ \ldots,\ \xi^{k-1}))).$$

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \xi^k.$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются предположения 1 и*

(*iii*) —

$$\rho_k \geqslant 0, \quad \sum_{k=1}^{\infty} \rho_k = \infty, \quad \sum_{k=1}^{\infty} \rho_k^2 \lambda_k < \infty;$$

(*iv*) — $\bar{\xi}^k$ *и* f_1 *удовлетворяют неравенствам*

$$E(\|\bar{\xi}^k\|^2/\xi^1,\ \ldots,\ \xi^{k-1}) \leqslant \lambda_k + \delta_1 f_1(x^{k-1}) + \\ + \delta_2(\nabla f_1(x^{k-1}),\ E(\bar{\xi}^k/\xi^1,\ \ldots,\ \xi^{k-1}));$$

(*v*) —

$$\sum_{k=1}^{\infty} \rho_k^2 < \infty \quad \text{или} \quad \lambda_k \equiv 0,\ \delta_1 = 0,\ \overline{\lim}\, \rho_k < 2/\gamma\delta_2.$$

Тогда при любом x^1 последовательность $\{x^k\}_{k=1}^{\infty}$, порожденная алгоритмом 1, почти наверное такова, что для нее существует предел последовательности $\{f_1(x^k)\}_{k=1}^{\infty}$ и

$$\underline{\lim}\,(\nabla f_1(x^{k-1}),\ E\bar{\xi}^k) \overset{\text{п.н}}{=} 0.$$

Теорема 1'. *Пусть в дополнение к условиям теоремы 1* (**vi**) —

$$(\nabla f_1(x^{k-1}),\ E(\bar{\xi}/\xi^1,\ \ldots,\ \xi^{k-1})) \geqslant \delta(\varepsilon) > 0 \text{ при } f_1(x^{k-1}) \geqslant f_1^* + \varepsilon$$

для всех $\varepsilon > 0$. *Тогда, почти наверное*

$$\lim_{k\to\infty} f_1(x^k) = f_1^*.$$

Теорема 1''. *Пусть в дополнение к условиям теоремы 1 выполнены условия* (**vii**) — *множества* $\{x \mid f_1(x) \leqslant \text{const}\}$ *ограничены;*
(**viii**) —

$$(\nabla f_1(x^{k-1}),\ E(\bar{\xi}^k/\xi^1,\ \ldots,\ \xi^{k-1})) \geqslant \delta(\varepsilon) > 0$$

при $\|\nabla f_1(x^{k-1})\| \geqslant \varepsilon$ *для всех* $\varepsilon > 0$.

Тогда почти наверное найдутся подпоследовательность $\{k_i\}_{i=1}^{\infty}$ *и точка* x^* *такие, что*

$$\nabla f_1(x^*) = 0;\quad \lim_{i\to\infty} x^{k_i} = x^*;\quad \lim_{k\to\infty} f_1(x^k) = f_1(x^*).$$

Теорема 1'''.. *Пусть в дополнение к условиям теоремы 1 множество* X^* *точек минимума функции* f_1 *непустое и*
(**ix**) —

$$\inf_x f_1(x) > f_1^* \text{ при } d_1(x,\ X^*) \geqslant \varepsilon;$$

$$d_1(x,\ X^*) = \inf_{y\in X^*} \|x - y\|,$$

(**x**) —

$$(\nabla f_1(x^{k-1}),\ E(\bar{\xi}^k/\xi^1,\ \ldots,\ \xi^{k-1})) \geqslant \delta(\varepsilon) > 0 \text{ при } d_1(x,\ X^*) \geqslant \varepsilon$$

для всех $\varepsilon > 0$.

Тогда

$$\lim_{k\to\infty} d_1(x^k,\ X^*) \overset{\text{п.н}}{=} 0;\quad \lim_{k\to\infty} f_1(x^k) \overset{\text{п.н}}{=} f_1^*.$$

В частности, если X^* *состоит из единственной точки* x^*, *то* $x^k \overset{\text{п.н.}}{\to} x^*$ *при* $k \to \infty$.

Библиографические указания. Параграф написан на основании работы [303].

3.6. Квазиградиентные методы

З а д а ч а 0. Найти $\arg\min_x f_0(x)$ для слабовыпуклой вниз функции $f_0 : R^n \to R^1$.

Определение 1. Функция f_0 называется слабовыпуклой вниз, если

для любого x из произвольного замкнутого ограниченного множества существует непустое множество $G(x)$ векторов $\tilde{\nabla} f_0(x)$ таких, что для всех z и $\tilde{\nabla} f_0(x) \in G(x)$

$$f_0(z) - f_0(x) \geqslant (\tilde{\nabla} f_0(x), z - x) + r(z, x) \qquad (3.5)$$

и $r(x, y) \| x - y \|^{-1} \to 0$ равномерно по x при $y \to x$.

Определение 2. Вектор $\tilde{\nabla} f_0(x)$, удовлетворяющий неравенству (3.5), называют *квазиградиентом* слабовыпуклой функции f_0 в точке x.

Слабовыпуклыми функциями являются дифференцируемые, а также выпуклые функции (не обязательно дифференцируемые). В первом случае квазиградиентом является обычный градиент, а во втором — обобщенный градиент. Класс слабовыпуклых функций оказывается замкнутым относительно операции взятия максимума, т. е. если $f(x, y)$ — слабовыпуклые при каждом значении y функции, то

$$f_0(x) \triangleq \max_{y \in Y} f(x, y) = f(x, y(x))$$

является слабовыпуклой функцией и

$$\tilde{\nabla} f_0(x) = \tilde{\nabla}_x f(x, y)\big|_{y = \arg\max_{y \in Y} f(x, y)}.$$

1. Квазиградиентный метод минимизации слабовыпуклой вниз функции

Задача 1. Найти $\arg\min_x f_0(x)$ для слабовыпуклой вниз функции $f_0 : R^n \to R^1$.

Множество решений X^* для этой задачи определим равенством $X^* = \{x \mid 0 \in G(x)\}$, где $G(x)$ — множество квазиградиентов функции f_0 в точке x.

Сущность данного метода заключается в построении в k-й итерации квазиградиента слабовыпуклой функции в точке x^k. Если движение в направлении квазиградиента выводит за пределы специальным образом построенного множества S, то процесс построения такой последовательности точек прерывается, и алгоритм начинает работать с произвольной точки некоторого подмножества $A \subset S$.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение x^0, постоянную $\delta > 0$, шаговый множитель ρ_0.

II. Построить множество S по правилу

$$S = Q(f_0(x^0) + \delta),$$

где $Q(\alpha) \triangleq \{x \mid f_0(x) \leqslant \alpha\}$.

III. Задать произвольное компактное подмножество A множества S.

IV. Положить $k = 0$.

Основной цикл. V. Вычислить квазиградиент $\tilde{\nabla} f_0 (x^k)$ функции f_0 в точке x^k.

VI. Вычислить вектор

$$x^{k+1} = \begin{cases} x^k - \rho_k \tilde{\nabla} f_0 (x^k), & \text{если } x^k + \rho_k \tilde{\nabla} f_0 (x^k) \in S, \\ y \in A, & \text{если } x^k + \rho^k \tilde{\nabla} f_0 (x^k) \notin S, \end{cases}$$

где y — произвольная точка множества A.

VII. Вычислить шаговый множитель ρ_{k+1}.

VIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если выполнены условия:* (i) — $X^* \triangleq \{x \mid 0 \in G(x)\}$ *компактно;* (ii) — $Q(\alpha) \triangleq \{x \mid f_0(x) \leqslant \alpha\}$ *компактно для любого* α; (iii) — *функция* f_0 *принимает на* X^* *конечное число значений;* (iv) — *шаговые множители удовлетворяют условиям*

$$\rho_k > 0; \quad \rho_k \to 0; \quad \rho_{k+1}/\rho_k \to 1; \quad \sum_{k=0}^{\infty} \rho_k = \infty,$$

то любая предельная точка последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, принадлежит множеству решений* X^* *задачи 1.*

2. Стохастический квазиградиентный метод минимизации слабовыпуклой функции

Алгоритм 2

Начало. I. Выбрать произвольную константу $\alpha \in (0, \infty)$ и произвольное замкнутое множество B, содержащееся в сфере радиуса α, т. е. $B \subset S \triangleq \{x \mid \|x\| \leqslant \alpha\}$, выбрать произвольную начальную точку $x^0 \in \boldsymbol{R}^n$.

II. Задать правило формирования последовательности шаговых множителей $\{\rho_k\}_{k=0}^{\infty}$.

III. Положить $k = 0$, найти ρ_0.

Основной цикл. IV. Вычислить случайный вектор ξ^k, условное математическое ожидание которого

$$E(\xi^k / x^0, \ldots, x^k) = \tilde{\nabla} f_0 (x^k) + b^k,$$

где b^k вектор, измеримый относительно σ-подалгебры $\mathcal{B}_k$, индуцированной случайными величинами $(x^0, x^1, \ldots, x^k)$; $\tilde{\nabla} f_0 (x^k)$ — квазиградиент слабовыпуклой функции f_0.

V. Вычислить вектор

$$x^{k+1} = \begin{cases} x^k - \rho_k \xi^k, & \text{если } \|x^k\| \leqslant \alpha, \\ z^{k+1} \in B, & \text{если } \|x^k\| > \alpha, \end{cases}$$

где z^{k+1} — произвольная точка множества B.

VI. Найти ρ_{k+1}.

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Пусть f_0 слабовыпуклая вниз функция и, кроме того, выполняются условия*

(i) —

$$\max_{x \in B} f_0(x) < \inf_{\|x\| > \alpha} f_0(x);$$

(ii) —

$$\|\xi^k\| + \|\tilde{\nabla} f_0(x^k)\| + \|b^k\| \leq \beta;$$

(iii) — *шаговые множители ρ_k, измеримые относительно σ-подалгебры $\mathcal{B}_k$ и таковы, что*

$$\rho_k \geq 0, \quad \rho_{k+1}/\rho_k \to 1, \quad \sum_{k=0}^{\infty} \rho_k = \infty;$$

$$\sum_{k=0}^{\infty} \rho_k \|b^k\| < \infty \text{ п. н.}, \quad \sum_{k=0}^{\infty} E\rho_k^2 < \infty.$$

Тогда почти при каждом ω предельные точки последовательности $\{x^k(\omega)\}_{k=0}^{\infty}$, порожденной алгоритмом 2, принадлежат множеству $X^* \triangleq \{x \mid 0 \in G(x)\}$, и последовательность $\{f_0(x^k(\omega))\}_{k=0}^{\infty}$ сходится ($G(x)$ — множество квазиградиентов функции f_0 в точке x).

Библиографические указания. Параграф написан на основании работ [267, 268, 269, 272, 153, 274].

3.7. ε-квазиградиентные методы

1. ε-квазиградиентный метод минимизации выпуклых функций

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$, $f_0 : R^n \to R^1$ — выпуклая функция.

За направление движения к следующему приближению x^{k+1} в данном методе выбирают ε-квазиградиент выпуклой функции f_0 в точке x^k.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, шаговый множитель ρ_0 и число ε_0, удовлетворяющее условиям теоремы 1, и некоторую константу $\alpha \geq 1$.

II. Положить $k = 0$.

Основной цикл. III. Положить $\varepsilon = \varepsilon_k$.

IV. Вычислить ε-квазиградиент $g_\varepsilon(x^k)$ функции f_0 в точке x^k.

V. Вычислить нормирующий множитель

$$\gamma_k = \begin{cases} 1, & \|g_\varepsilon(x^k)\| \leq \alpha, \\ \|g_\varepsilon(x^k)\|^{-2}, & \|g_\varepsilon(x^k)\| > \alpha. \end{cases}$$

VI. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \gamma_k g_\varepsilon(x^k).$$

VII. Вычислить шаговый множитель ρ_{k+1} и число ε_{k+1}, удовлетворяющие условиям теоремы 1.

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть функция f_0 выпукла и множество $X^* \triangleq \{x \mid 0 \in G(x, 0)\}$ (здесь $G(x, \varepsilon)$ ($\varepsilon \geqslant 0$) — множество ε-квазиградиентов функции f_0 в точке x) компактно. Тогда, если числовые последовательности $\{\rho_k\}_{k=0}^{\infty}$* и *$\{\varepsilon_k\}_{k=0}^{\infty}$ таковы, что*

$$\rho_k \to +0,\ \varepsilon_k \to +0 \text{ при } k \to \infty \text{ и } \sum_{k=0}^{\infty} \rho_k = \infty,$$

то предел любой сходящейся подпоследовательности последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, принадлежит множеству X^*.

Теорема 1′. *Пусть функция f_0 удовлетворяет условиям теоремы 1 и $\varepsilon_k = \text{const} > 0$, $k = 0, 1, \ldots$. Тогда, если числовая последовательность $\{\rho_k\}_{k=0}^{\infty}$ такова, что*

$$\rho_k \to +0 \text{ при } k \to \infty,\ \sum_{k=0}^{\infty} \rho_k = \infty,$$

то существует подпоследовательность $\{x^{k_s}\}_{s=0}^{\infty}$ последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, сходящаяся к множеству

$$X_\varepsilon^* \triangleq \{x \mid f_0(x) \leqslant \min_{x'} f_0(x') + \varepsilon\}.$$

2. ε-квазиградиентный метод минимизации слабовыпуклых функций

Задача 2. Найти $\arg\min_x f_0(x)$ для заданной слабовыпуклой функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$.

В данном итеративном методе последующее $(k+1)$-е приближение x^{k+1} выбирают в направлении ε-квазиградиента $g_\varepsilon(x^k)$ для функции f_0 в точке x^k (k-м приближении).

Определение 2. Вектор $g_\varepsilon(x)$ называют ε-квазиградиентом слабовыпуклой функции f_0 в точке x ($\varepsilon > 0$), если для всех $z \in \boldsymbol{R}^n$ выполняется неравенство

$$f_0(z) - f_0(x) \geqslant (g_\varepsilon(x),\ z - x) + r(x, z) - \varepsilon$$

и отношение $\frac{r(x,z)}{\|z - x\|}$ стремится к нулю равномерно по x при $z \to x$ (в каждом компактном подмножестве из $\boldsymbol{R}^n$).

Если функция f_0 определяется равенством

$$f_0(x) = \max_{y \in Y} \varphi(x, y),$$

где множество Y компактно, а $\varphi(x, y)$ непрерывна по совокупности переменных и равномерно слабовыпуклая по x для каждого y,

то ε-квазиградиентом функции f_0 в точке $\bar{x}$ является квазиградиент функции $\varphi(x, \bar{y})$ по x в точке $x = \bar{x}$, где $\bar{y}$ — произвольный вектор из множества

$$Y_\varepsilon^*(\bar{x}) \triangleq \{y^* \mid \varphi(\bar{x}, y^*) \geqslant \varphi(\bar{x}, y) - \varepsilon, \ \forall y \in Y\}.$$

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, некоторую постоянную $\delta > 0$, шаговый множитель ρ_0 и величину ε_0, удовлетворяющие условиям теоремы 2.

II. Положить $k = 0$.

Основной цикл. III. Если $f_0(x^k) > f_0(x^0) + \delta$, то положить $x^{k+1} = x^0$ и перейти к шагу VI; иначе перейти к шагу IV.

IV. Вычислить ε_k-квазиградиент $g_{\varepsilon_k}(x^k)$ функции f_0 в точке x^k.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k g_{\varepsilon_k}(x^k).$$

VI. Вычислить шаговый множитель ρ_{k+1} и величину ε_{k+1}, удовлетворяющие условиям теоремы 2.

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть функция f_0 слабовыпуклая и выполнены условия: (i) — множества $\{x \mid f_0(x) \leqslant \alpha\}$ компактны при всех α; (ii) — множество $X^* \triangleq \{x \mid 0 \in G(x, 0)\}$ компактно; (iii) — числовые последовательности $\{\rho_k\}_{k=0}^{\infty}$, $\{\varepsilon_k\}_{k=0}^{\infty}$ таковы, что*

$$\rho_k \to +0, \ \rho_{k+1}/\rho_k \to 1, \quad \varepsilon_k \to +0$$

при $k \to \infty$, $\sum_{k=0}^{\infty} \rho_k = \infty$, $\sum_{k=0}^{\infty} \varepsilon_k < \infty$. Тогда предел любой сходящейся подпоследовательности последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, принадлежит множеству X^*.

Замечание 2. Чтобы избежать вычисления на каждой итерации значения функции $f_0(x^k)$, шаг III алгоритма заменяют на III'.

III'. Если $\| x^k \| > \beta$, то положить $x^{k+1} = x^0$ и перейти к шагу VI; иначе перейти к шагу IV (здесь константу β выбирают из условия $f_0(x) > f_0(x^0) + \delta$ при $\| x \| > \beta$).

Библиографические указания. При написании параграфа использованы работы [275, 272, 204].

3.8. Методы обобщенных почти градиентов для минимизации функций, удовлетворяющих локальному условию Липшица

1. Детерминированный случай

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для функции $f_0 : R^n \to R^1$, удовлетворяющей локальному условию Липшица.

Если существует практически эффективный способ вычисления обобщенного почти градиента $\hat{g}(x)$ функции $f_0(x)$ в любой точке $x \in R^n$, то за направление движения к следующему приближению x^{k+1} в приводимых ниже алгоритмах выбирается вектор ξ^k, равный обобщенному почти градиенту функции f_0 в точке $\tilde{x}^k$, равномерно распределенной в n-мерном кубе с центром в точке x^k со стороной $\alpha_k > 0$, т. е.

$$\xi^k = \hat{g}(\tilde{x}^k). \qquad (3.6)$$

В других случаях за направление движения ξ^k выбирается конечно-разностный аналог обобщенного почти градиента $\hat{g}(\tilde{x}^k)$

$$\xi^k = \frac{1}{\alpha_k} \sum_{i=1}^{n} \left(f_0\left(\tilde{x}_1^k, \ldots, x_i^k + \frac{\alpha_k}{2}, \ldots, \tilde{x}_n^k\right) - \right.$$
$$\left. - f_0\left(\tilde{x}_1^k, \ldots, x_i^k - \frac{\alpha_k}{2}, \ldots, \tilde{x}_n^k\right)\right) e^i, \qquad (3.7)$$

где e^i, $i = 1, \ldots, n$ — i-й орт.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, шаговый множитель ρ_0 и величину смещения α_0, удовлетворяющие условиям теоремы 1.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализацию $\tilde{x}^k$ случайной точки, равномерно распределенной в n-мерном кубе с центром в точке x^k со стороной α_k.

IV. Вычислить вектор ξ^k направления движения по формуле (3.6) или (3.7).

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \xi^k.$$

VI. Вычислить значение шагового множителя ρ_{k+1} и величину смещения α_{k+1}, удовлетворяющие условиям теоремы 1.

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть функция f_0 в любой ограниченной области удовлетворяет условию Липшица и, кроме того, выполняются условия*

$$\rho_k > 0, \quad \alpha_k > 0, \quad k = 0, 1, \ldots;$$
$$\sum_{k=0}^{\infty} \rho_k = \infty; \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty; \quad \rho_k/\alpha_k \to 0,$$
$$(\alpha_k - \alpha_{k+1})/\rho_k \to 0, \quad \alpha_k \to 0$$

при $k \to \infty$.

Тогда, если последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, принадлежит ограниченному множеству пространства R^n,

то с вероятностью единица предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$ *принадлежат множеству* $X^* \triangleq \{x^* \mid 0 \in G(x^*)\}$ *и последовательность* $\{f_0(x^k)\}_{k=0}^{\infty}$ *почти наверное сходится.*

Здесь и далее $G(x)$ — множество обобщенных почти градиентов функции f_0 в точке x.

2. Стохастический случай

Задача 2. Найти $\arg\min_x \boldsymbol{E} f_0(x, \omega)$ для заданной функции $f_0 : \boldsymbol{R}^n \times \Omega \to \boldsymbol{R}^1$.

Предположение 2. Функция f_0 удовлетворяет локальному условию Липшица

$$|f_0(x, \omega) - f_0(y, \omega)| \leqslant \gamma(\omega) \|x - y\|, \quad E\gamma^2(\omega) < \infty.$$

Если существует достаточно эффективный способ построения обобщенного почти градиента $\hat{g}(x, \omega)$ функции f_0 в любой точке (x, ω), то в приводимом ниже алгоритме за направление движения ξ^k к следующему приближению x^{k+1} выбирается вектор

$$\xi^k = \hat{g}(\tilde{x}^k, \omega^k), \tag{3.8}$$

где $\tilde{x}^k$ — реализация случайного вектора, равномерно распределенного в n-мерном кубе с центром в точке x^k и стороной $\alpha_k > 0$; ω^k — независимые наблюдения ω. Если же построение $\hat{g}(x, \omega)$ затруднительно, то за направление движения ξ^k выбирается конечно-разностный аналог обобщенного почти градиента $\hat{g}(\tilde{x}^k, \omega^k)$ в виде

$$\xi^k = \frac{1}{2\alpha_k} \sum_{i=1}^{n} (f_0(\tilde{x}_1^k, \ldots, x_i^k + \alpha_k, \ldots, \tilde{x}_n^k, \omega^k) - \\ - f_0(\tilde{x}_1^k, \ldots, x_i^k - \alpha_k, \ldots, \tilde{x}_n^k, \omega^k)) e^i, \tag{3.9}$$

где e^i, $i = 1, \ldots, n$ — i-й орт.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, значения шагового множителя ρ_0 и смещения α_0, удовлетворяющие условиям теоремы 2; положить $k = 0$.

Основной цикл. II. Вычислить реализации $\tilde{x}_i^k$, $i = 1, \ldots, n$, случайных величин, равномерно распределенных на отрезках

$$[x_i^k - \alpha_k, \ x_i^k + \alpha_k], \quad i = 1, \ldots, n.$$

III. Вычислить реализацию ω^k случайного параметра ω.

IV. Вычислить вектор ξ^k направления движения к следующему приближению x^{k+1} по формуле (3.8) (или (3.9)).

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \xi^k.$$

VI. Вычислить значение шагового множителя ρ_{k+1} и смещения α_{k+1}, удовлетворяющие условиям теоремы 2.

VII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 2. *Пусть выполнены предположение 2 и условия*

$$\rho_k > 0, \quad \alpha_k > 0 \quad \text{при} \quad k = 0, 1, \ldots;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty, \quad \sum_{k=0}^{\infty} (\rho_k/\alpha_k)^2 < \infty;$$

$$\alpha_k \to 0 \text{ и } |\alpha_k - \alpha_{k+1}|/\rho_k \to 0 \text{ при } k \to \infty.$$

Тогда существует подпоследовательность $\{x^{k_s}\}_{s=0}^{\infty}$ *последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 2, стремящаяся к точке* $x^* \in X^* \triangleq \{x^* \mid 0 \in G(x^*)\}$.

Библиографические указания. При написании параграфа использованы работы [93, 97, 102, 100, 101].

Дополнительные сведения о минимизации функций, удовлетворяющих условию Липшица, можно найти в работе [483].

3.9. Метод усреднения направлений спуска для минимизации функций, удовлетворяющих условию Липшица

Задача 1. Найти $\arg\min_{x \in R^n} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$.

Предположение 1. Функция f_0 удовлетворяет локальному условию Липшица.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать произвольное натуральное число l.

III. Положить $k = 0$.

Основной цикл. IV. Найти шаговый множитель ρ_k, параметры α_k и δ_k, удовлетворяющие условиям теоремы 1.

V. Вычислить реализацию $\tilde{x}^k$ случайной точки, равномерно распределенной в n-мерном кубе с центром в точке x^k и стороной α_k.

VI. Вычислить вектор θ^k по формуле

$$\theta^k = \sum_{i=1}^{n} \frac{f_0(\tilde{x}^k + \delta_k e^i) - f_0(\tilde{x}^k)}{\delta_k} e^i,$$

где e^i, $i = 1, \ldots, n$, i-й орт.

VII. Если $k \geqslant l$, то вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k \sum_{j=k-l}^{k} \theta^j$$

и перейти к шагу VIII; иначе вычислить следующее приближение x^{k+1} по формуле

$$x^{k+1} = x^k - \rho_k \sum_{j=0}^{k} \theta^j$$

и перейти к шагу VIII.

VIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Если функция f_0 удовлетворяет локальному условию Липшица и выполняются условия*

$$\rho_k > 0, \quad \alpha_k > 0, \quad k = 0, 1, \ldots, ;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty, \quad \lim_{k\to\infty} (\rho_k/\alpha_k) = 0;$$

$$\lim_{k\to\infty} \alpha_k = 0, \quad \lim_{k\to\infty} (\delta_k/\alpha_k) = 0, \quad \overline{\lim_{k\to\infty}} (\rho_k/\rho_{k+1}) < \infty;$$

$$\lim_{k\to\infty} (|\alpha_k - \alpha_{k+1}| / \rho_k) = 0, \quad \lim_{k\to\infty} (\alpha_{k+1}/\alpha_k) = 1,$$

то все предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, с вероятностью 1 принадлежат множеству $X^* \triangleq \{x^* \mid 0 \in G(x^*)\}$, где $G(x)$ — множество обобщенных почти градиентов функции f_0 в точке x, и, кроме того, последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ сходится почти наверное.

Библиографические указания. При написании параграфа использовались работы [96, 97].

3.10. Конечно-разностный метод минимизации разрывных функций

Задача 1. Найти $\arg\min_{x\in R^n} f_0(x)$ для разрывной функции $f_0 : R^n \to R^1$.

Предположения 1. Функция f_0: (i) непрерывная и дифференцируемая почти всюду; (ii) ограничена в любой ограниченной области R^n; (iii) полунепрерывная снизу, т. е. если $\{x^k\}_{k=0}^{\infty}$ — произвольная последовательность точек, сходящихся к $\bar{x}$, и предел последовательности $\{f_0(x^k)\}_{k=0}^{\infty}$ существует, то $\lim_{k\to\infty} f_0(x^k) \geqslant f_0(\bar{x})$.

Определение 1. Точка x^* является (квази) решением задачи 1, если $x^* \in X^* \triangleq \{x \mid 0 \in \text{co}\, S(x)\}$, где $S(x)$ — множество предельных точек последовательности векторов, j-я компонента которых

$$\frac{1}{\varepsilon_k}\left(f_0\left(x_1^k, \ldots, x_j^k + \frac{\varepsilon_k}{2}, \ldots, x_n^k\right) - f_0\left(x_1^k, \ldots, x_j^k - \frac{\varepsilon_k}{2}, \ldots, x_n^k\right)\right),$$

здесь $\{x^k\}_{k=0}^{\infty}$ — произвольная последовательность точек, сходящаяся к x, а $\{\varepsilon_k\}_{k=0}^{\infty}$ — произвольная последовательность, сходящаяся к 0.

Основная идея метода состоит в том, что разрывная и негладкая функция $f_0(\cdot)$ приближается последовательностью непрерывных сглаженных функций $\bar{f}(\cdot, k)$, которые, за исключением точек разрыва, сходятся к $f_0(\cdot)$ при $k \to \infty$. За направление движения к следующему приближению x^{k+1} в k-й итерации выбирается случайный вектор, который является статистической оценкой градиента сглаженной функции.

Функцию $\bar{f}(\cdot, k)$ можно определить как

$$\bar{f}(x, k) = \frac{1}{\alpha_k^n} \int\limits_{x_1-\frac{\alpha_k}{2}}^{x_1+\frac{\alpha_k}{2}} \cdots \int\limits_{x_n-\frac{\alpha_k}{2}}^{x_n+\frac{\alpha_k}{2}} \tilde{f}(y, k)\, dy_1, \ldots, dy_n,$$

где $\{\alpha_k\}_{k=0}^{\infty}$ — последовательность положительных чисел, стремящихся к нулю; $\tilde{f}(y, k)$ определяется по правилу

$$\tilde{f}(y, k) = \frac{1}{\alpha_k^n} \int\limits_{y_1-\frac{\alpha_k}{2}}^{y_1+\frac{\alpha_k}{2}} \cdots \int\limits_{y_n-\frac{\alpha_k}{2}}^{y_n+\frac{\alpha_k}{2}} f_0(t_1, \ldots, t_n)\, dt_1, \ldots, dt_n.$$

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, шаговый множитель ρ_0 и смещение α_0, удовлетворяющие условиям теоремы 1.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализации τ_i^k, $i = 1, \ldots, n$, независимых случайных величин τ_i, $i = 1, \ldots, n$, равномерно распределенных на отрезках $[-\alpha_k/2, \alpha_k/2]$.

IV. Вычислить

$$\xi_i^k = \bar{\xi}_i^k + \nu_i^k, \quad i = 1, \ldots, n,$$

где $\bar{\xi}_i^k$, ν_i^k, $i = 1, \ldots, n$ — соответственно реализации независимых случайных величин $\bar{\xi}_i$, ν_i, $i = 1, \ldots, n$, равномерно распределенных на отрезках $[-\alpha_k/2, \alpha_k/2]$.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \frac{\rho_k}{\alpha_k} \sum_{i=1}^{n} \Big(f_0\Big(x_1^k + \xi_1^k, \ldots, x_i^k + \tau_i^k + \frac{\alpha_k}{2}, \ldots, x_n^k + \xi_n^k\Big) - f_0\Big(x_1^k + \xi_1^k, \ldots, x_i^k + \tau_i^k - \frac{\alpha_k}{2}, \ldots, x_n^k + \xi_n^k\Big)\Big) e^i,$$

где e^i, $i = 1, \ldots, n$ — i-й орт.

VI. Вычислить значение шагового множителя ρ_{k+1} и значение смещения α_{k+1}, удовлетворяющие условиям теоремы 1.

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполнены предположения 1 и, кроме того, имеют место условия*

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} (\rho_k/\alpha_k^2)^2 < \infty;$$

$$\frac{|\alpha_k - \alpha_{k+1}|}{\alpha_k \rho_k} \to 0, \quad \alpha_k \to 0 \quad \text{при} \quad k \to \infty.$$

Тогда с вероятностью 1 существует подпоследовательность $\{x^{k_m}\}_{m=0}^{\infty}$ *последовательности* $\{x^k\}_{k=0}^{\infty}$*, порожденной алгоритмом 1, стремящаяся к точке* $x^* \in X^*$*, для которой*

$$\lim_{k_m \to \infty} \nabla \bar{f}(x^{k_m}, k_m) = 0.$$

Библиографические указания. При написании параграфа использованы работы [103, 97].

3.11. Метод линеаризации решения дискретных минимаксных задач

З а д а ч а 1. Найти $\arg\min_{x \in R^n} f_0(x)$, где

$$f_0(x) \triangleq \max_{1 \leqslant i \leqslant m} f_i(x),$$

здесь $f_i(x)$ — заданные непрерывно дифференцируемые функции.

Обозначим через $\mathcal{I}_\delta(x)$ множество

$$\{i \mid 1 \leqslant i \leqslant m, \ f_i(x) \geqslant f_0(x) - \delta\}, \quad \delta \geqslant 0.$$

Алгоритм 1

Н а ч а л о. I. Выбрать произвольное начальное приближение $x^0 \in R^n$ и константы $\varepsilon \in (^1/_2, 1)$, $\delta > 0$ (рекомендуется выбирать δ достаточно малым; $\varepsilon = {}^2/_3$); положить $k = 0$.

О с н о в н о й ц и к л. II. Положить $x = x^k$.

III. Найти $p(x)$ и $\beta(x)$ — решение следующей задачи выпуклого программирования: найти

$$\arg\min_{\beta, p} \left(\beta + \frac{1}{2} \| p \|^2\right)$$

при ограничениях

$$(\nabla f_i(x), p) + f_i(x) - \beta \leqslant 0, \quad i \in \mathcal{I}_\delta(x).$$

IV. Если $p(x) = 0$, то положить $x^* = x$ и прекратить вычисления; иначе перейти к шагу V.

V. Положить $\tau = 0$.

VI. Положить $\alpha_k = (^1/_2)^\tau$.

VII. Если выполняется неравенство

$$f_0(x + \alpha_k p(x)) \leqslant f_0(x) - \alpha_k \varepsilon \| p(x) \|^2,$$

то перейти к шагу VIII; иначе положить $\tau = \tau + 1$ и перейти к шагу VI.

VIII. Положить $x^{k+1} = x + \alpha_k p(x)$, положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Пусть $f_i(x)$, $i = 1, \ldots, m$, непрерывно дифференцируемые функции; область $X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0)\}$ ограничена и $\nabla f_i(x)$, $i = 1, \ldots, m$, удовлетворяют в X_0 условию Липшица с константой $\gamma < \infty$. Тогда любая предельная точка x^* бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, удовлетворяет необходимым условиям минимума $f_0(x)$ при $x \in R^n$. Если, кроме того, $f_i(x)$, $i \in \overline{1, m}$ — выпуклы, то x^* — решение задачи 1.*

Необходимым условием минимума $f_0(x)$ в точке x^* является существование чисел $\bar{u}_i$, $i = 1, \ldots, m$, таких, что

$$\sum_{i=1}^{m} \bar{u}_i \nabla f_i(x^*) = 0;$$

$$\bar{u}_i (f_i(x^*) - f_0(x^*)) = 0, \quad i \in \overline{1, m};$$

$$\sum_{i=1}^{m} \bar{u}_i = 1, \quad \bar{u}_i \geqslant 0, \quad i = 1, \ldots, m.$$

Следующая теорема дает локальную оценку сходимости алгоритма 1.

Теорема 1'. *Пусть x^* — точка минимума $f_0(x)$, функции $f_i(x)$, $i \in \mathcal{I}_0(x^*)$ — дважды непрерывно дифференцируемы. Кроме того, пусть градиенты $\nabla f_i(x^*)$, $i \in \mathcal{I}_0(x^*)$, где*

$$\mathcal{I}_0(x^*) \triangleq \{i \mid i \in \overline{1, m},\ f_i(x^*) = f_0(x^*)\},$$

таковы, что разности

$$\nabla f_i(x^*) - \nabla f_{i_0}(x^*), \quad i \neq i_0, \quad i_0 \in \mathcal{I}_0(x^*),$$

линейно-независимы и множители $\bar{u}_i$ строго больше нуля для $i \in \mathcal{I}_0(x^*)$, а $(y, \nabla^2_{xx}\varphi(x^*, \bar{u})\, y) > 0$ для всех $y \neq 0$, здесь

$$\varphi(x, u) = \sum_{i=1}^{m} u_i f_i(x),$$

а $\nabla^2_{xx}\varphi(x, u)$ — матрица вторых производных относительно x.

Тогда при достаточно малом $\delta > 0$ и $\alpha > 0$ существует такая окрестность точки x^*, что алгоритм 1 сходится с любого начального приближения x^0 из этой окрестности и

$$\|x^* - x^k\| \leqslant c_1 q^k, \quad 0 < q < 1.$$

Теорема 1''. *Пусть выполнены все условия теоремы 1', и, кроме того, число индексов в множестве $\mathcal{I}_0(x^*)$ равно $n + 1$.*

В этом случае при достаточно малом $\delta > 0$ алгоритм 1 в окрестности x^* сходится при постоянном $\alpha_k \equiv 1$ с квадратичной скоростью к точке x^*.

Библиографические указания. Параграф написан на основании работы [320]

3.12. Методы последовательных приближений для решения дискретных минимаксных задач

Задача 0. Найти $\arg \min\limits_{x \in R^n} \max\limits_{i \in \mathcal{I}} \varphi_i(x)$ для заданных функций $\varphi_i : R^n \to R^1$, $i \in \mathcal{I}$ и заданного множества $\mathcal{I}$.

Предположение 0. (*i*) — функции φ_i, $i \in \mathcal{I}$,— непрерывно дифференцируемы в R^n.

В данном параграфе описываются методы, в которых на k-й итерации в качестве направления движения к следующему приближению x^{k+1} выбирается вектор $h^k(\varepsilon)$, удовлетворяющий условию

$$\max_{i \in \mathcal{K}_\varepsilon(x^k)} (\nabla\varphi_i(x^k), h^k(\varepsilon)) = \min_{\|h\|=1} \max_{i \in \mathcal{K}_\varepsilon(x^k)} (\nabla\varphi_i(x^k), h).$$

где

$$\mathcal{K}_\varepsilon(x^k) \triangleq \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \ i \in \mathcal{I}\}.$$

Вектор $h^k(\varepsilon)$ при $\varepsilon \geqslant 0$ называют направлением ε-наискорейшего спуска.

1. Первый метод последовательных приближений

Алгоритм 1.

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать константы $\varepsilon_0 > 0$ и $\alpha_0 > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Положить $\varepsilon = 0$ и перейти к шагу VII.

V. Положить $j = 0$.

VI. Положить $\varepsilon = \varepsilon_j$.

VII. Найти множество индексов

$$\mathcal{K}_\varepsilon(x^k) = \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \ i \in \mathcal{I}\}. \tag{3.10}$$

VIII. Найти многогранник $L_\varepsilon(x^k)$, который является выпуклой оболочкой, натянутой на множество точек

$$\{\nabla\varphi_i(x^k), \ i \in \mathcal{K}_\varepsilon(x^k)\}.$$

IX. Если $\varepsilon = 0$, то перейти к шагу X; иначе перейти к шагу XI.

X. Используя алгоритм 1′, определить, принадлежит ли начало координат многограннику $L_0(x^k)$. Если начало координат

принадлежит $L_0(x^k)$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу XI.

XI. Используя алгоритм 1″, найти точку z_ε, которая является ближайшей к началу координат точкой многогранника $L_\varepsilon(x^k)$.

XII. Вычислить $\psi_\varepsilon(x^k)$ по формуле $\psi_\varepsilon(x^k) = -\|z_\varepsilon\|$.

XIII. Если выполняется неравенство $\psi_\varepsilon(x^k) \leqslant -\varepsilon(\alpha_0/\varepsilon_0)$, то перейти к шагу XIV; иначе положить $\varepsilon_{j+1} = \varepsilon_0/2^{j+1}$, $j = j+1$ и перейти к шагу VI.

XIV. Вычислить вектор $h^k(\varepsilon)$ — направление ε-наискорейшего спуска функции $\max_{i\in\mathcal{J}} \varphi_i(x)$ в точке x^k по формуле $h^k(\varepsilon) = -(1/\|z_\varepsilon\|)z_\varepsilon$.

XV. Вычислить шаговый множитель ρ_k из условия

$$\max_{i\in\mathcal{J}} \varphi_i(x^k + \rho_k h^k(\varepsilon)) = \min_{\rho\geqslant 0} \max_{i\in\mathcal{J}} \varphi_i(x^k + \rho h^k(\varepsilon)).$$

XVI. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k(\varepsilon).$$

XVII. Положить $k = k+1$ и перейти к шагу IV.

Теорема 1. *Если выполнено предположение 0 и начальное приближение x^0 в алгоритме 1 таково, что множество*

$$X_0 \triangleq \{x \mid \max_{i\in\mathcal{J}} \varphi_i(x) \leqslant \max_{i\in\mathcal{J}} \varphi_i(x^0), \quad x \in \boldsymbol{R}^n\},$$

ограничено, то любая предельная точка бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, является стационарной точкой функции $\max_{i\in\mathcal{J}} \varphi_i(x)$.

(Точка $x^* \in \boldsymbol{R}^n$, для которой выполняется неравенство

$$\inf_{\|g\|=1} \max_{i\in\mathcal{K}_0(x^*)} \left(\frac{\partial\varphi_i(x^*)}{\partial x}, g\right) \geqslant 0,$$

называется стационарной точкой функции $\max_{i\in\mathcal{J}} \varphi_i(x)$ на $\boldsymbol{R}^n$. Если функция $\max_{i\in\mathcal{J}} \varphi_i(x)$ выпукла, то стационарная точка x^* является точкой минимума).

Замечание 1. Чтобы на k-й итерации уменьшить количество сравнений $\psi_\varepsilon(x^k)$ с $-\varepsilon(\alpha_0/\varepsilon_0)$ в [127], рекомендуется начинать сравнение не с $\varepsilon = \varepsilon_0$, а со значения ε, полученного на предыдущей $(k-1)$-й итерации, т. е. при таком ε, которое получается при выходе из цикла, образованного шагами VI—XIII на $(k-1)$-й итерации.

Алгоритм 1′ **(алгоритм определения принадлежности начала координат многограннику $L_\varepsilon(x^k)$, который является выпуклой оболочкой множества векторов $\{\nabla\varphi_i(x^k),\ i \in \mathcal{K}_\varepsilon(x^k)\}$).**

I. Вычислить величину l — количество элементов множества $\mathcal{K}_\varepsilon(x^k)$.

II. Вычислить i_0 — наименьший элемент множества $\mathcal{K}_\varepsilon(x^k)$.

III. Положить $j = 1$.

IV. Положить $m = 0$.

V. Вычислить индекс $i = i_0 + m$.

VI. Если индекс i принадлежит множеству $\mathcal{K}_\varepsilon(x^k)$, то положить $g^j = \nabla\varphi_i(x^k)$ и перейти к шагу VII; иначе положить $m = m + 1$ и перейти к шагу V.

VII. Если $j < l$, то положить $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу VIII.

VIII. Решить задачу линейного программирования в $(l+1)$-мерном пространстве векторов $(\gamma_1, \gamma_2, \ldots, \gamma_l, \alpha)$: найти $\arg\min\alpha$ при условиях

$$\sum_{j=1}^{l}\gamma_j g_s^j - \alpha \leqslant 0, \quad s = 1, 2, \ldots, n;$$

$$-\sum_{j=1}^{l}\gamma_j g_s^j - \alpha \leqslant 0, \quad s = 1, 2, \ldots, n;$$

$$\sum_{j=1}^{l}\gamma_j = 1; \quad \gamma_j \geqslant 0, \quad j = 1, \ldots, l; \quad \alpha \geqslant 0.$$

Обозначить через $(\gamma_1^0, \gamma_2^0, \ldots, \gamma_l^0, \alpha^0)$ решение этой задачи.

IX. Если $\alpha^0 = 0$, то начало координат принадлежит многограннику $L_\varepsilon(x^k)$, если $\alpha^0 > 0$, то начало координат не принадлежит многограннику $L_\varepsilon(x^k)$.

Алгоритм 1″ **(алгоритм вычисления ближайшей к началу координат точки многогранника $L_\varepsilon(x^k)$, который является выпуклой оболочкой множества векторов** $Q = \{\nabla\varphi_i(x^k),\ i \in \mathcal{K}_\varepsilon(x^k)\}$).

I. В качестве начального приближения y^0 выбрать произвольную точку многогранника $L_\varepsilon(x^k)$ (в частности, в качестве y^0 можно взять вектор $\nabla\varphi_{i_0}(x^k)$, для которого

$$(\nabla\varphi_{i_0}(x^k),\ \nabla\varphi_{i_0}(x^k)) = \min_{i\in\mathcal{K}_\varepsilon(x^k)}(\nabla\varphi_i(x^k),\ \nabla\varphi_i(x^k)),$$

или вектор, равный $\sum_{j=1}^{l}\gamma_j^0 g^j$, где $(\gamma_1^0, \gamma_2^0, \ldots, \gamma_l^0)$ — вектор, полученный при решении задачи линейного программирования на шаге VIII алгоритма 1′, а вектора g^j, $j = 1, \ldots, l$, определяются шагами I—VII алгоритма 1′).

II. Положить $m = 0$.

III. Найти точку $\bar{y}^m$ множества Q, для которой

$$(\bar{y}^m,\ y^m) = \min_{i\in\mathcal{K}_\varepsilon(x^k)}(\nabla\varphi_i(x^k),\ y^m).$$

IV. Если $(y^m - \bar{y}^m, y^m) = 0$, то положить $z_\varepsilon = y^m$ и прекратить вычисления; иначе перейти к шагу V.

V. Найти параметр $\tau_m \in [0, 1]$, удовлетворяющий условию

$$\| y^m + \tau_m (\bar{y}^m - y^m) \| = \min_{t \in [0,1]} \| y^m + t (\bar{y}^m - y^m) \|.$$

VI. Вычислить следующее приближение

$$y^{m+1} = y^m + \tau_m (\bar{y}^m - y^m).$$

VII. Положить $m = m + 1$ и перейти к шагу III.

Теорема 1'. *Если выполнены предположения теоремы 1, то бесконечная последовательность* $\{y^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1'', сходится к ближайшей к началу координат точке* z_ε *многогранника* $L_\varepsilon (x^k)$.

2. Модификация первого метода последовательных приближений

Алгоритм 2

Все шаги алгоритма 1, за исключением шага XV, остаются без изменений. Шаг XV записать в виде:

XV. Используя алгоритм 2', вычислить шаговый множитель ρ_k, удовлетворяющий условию

$$(\max_{i \in \mathcal{I}} \varphi_i (x^k) - \max_{i \in \mathcal{I}} \varphi_i (x^k + \rho_k h^k (\varepsilon)))/(\max_{i \in \mathcal{I}} \varphi_i (x^k) - \\ - \min_{\rho \geqslant 0} \max_{i \in \mathcal{I}} \varphi_i (x^k + \rho h^k (\varepsilon))) \geqslant \beta, \quad (3.11)$$

где $\beta \in (0, 1)$ — произвольный параметр, фиксированный для всех значений k.

Алгоритм 2' **(алгоритм вычисления за конечное число итераций шагового множителя ρ_k, удовлетворяющего условию (3.11))**

I. Выбрать произвольный параметр $\beta \in (0, 1)$ и константу $\delta_0 > 0$.

II. Вычислить $\mu = (\sqrt{5} - 1)/2$.

III. Определить функцию $f_k : \boldsymbol{R}^1 \to \boldsymbol{R}^1$ по правилу

$$f_k (t) = \max_{i \in \mathcal{I}} \varphi_i (x^k + t h^k (\varepsilon)). \quad (3.12)$$

IV. Вычислить значения $f_k (\delta_0)$ и $f_k (0)$. Если $f_k (\delta_0) < f_k (0)$, то перейти к шагу V; иначе перейти к шагу X.

V. Положить $s = 1$.

VI. Вычислить $\delta_s = \delta_{s-1} \mu$.

VII. Если $f_k (\delta_s) \geqslant f_k (\delta_{s-1})$, то перейти к шагу IX; иначе перейти к шагу VIII;

VIII. Положить $s = s + 1$ и перейти к шагу VI.

IX. Положить $\delta_0^{(1)} = 0$, $\delta_0^{(2)} = \delta_{s-1}$, $\delta_0^{(3)} = \delta_s$ и перейти к шагу XV.

X. Положить $s = 1$.

XI. Вычислить $\delta_s = \delta_{s-1}/\mu$.

XII. Если $f_k(\delta_s) \leqslant f_k(0)$, то перейти к шагу XIV; иначе перейти к шагу XIII.

XIII. Положить $s = s + 1$ и перейти к шагу XI.

XIV. Положить $\delta_0^{(1)} = 0$, $\delta_0^{(2)} = \delta_s$, $\delta_0^{(3)} = \delta_{s-1}$.

XV. Положить $j = 0$.

XVI. Найти множество индексов $\mathcal{K} \subset \mathcal{I}$

$$\mathcal{K} \triangleq \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k + \delta_j^{(2)} h^k(\varepsilon)) - \varphi_i(x^k + \delta_j^{(2)} h^k(\varepsilon)) \leqslant 0\}.$$

Вычислить производную справа $f'_k(\delta_j^{(2)} + 0)$ и производную слева $f'_k(\delta_j^{(2)} - 0)$ функции f_k в точке $\delta_j^{(2)}$:

$$f'_k(\delta_j^{(2)} + 0) = \max_{i \in \mathcal{K}} (\nabla\varphi_i(x^k + \delta_j^{(2)} h^k(\varepsilon)),\ h^k(\varepsilon));$$

$$f'_k(\delta_j^{(2)} - 0) = \min_{i \in \mathcal{K}} (\nabla\varphi_i(x^k + \delta_j^{(2)} h^k(\varepsilon)),\ h^k(\varepsilon)).$$

XVII. Вычислить

$$\Delta_j = \max\{f'_k(\delta_j^{(2)} + 0)(\delta_j^{(2)} - \delta_j^{(3)}),\ f'_k(\delta_j^{(2)} - 0)(\delta_j^{(2)} - \delta_j^{(1)}),\ 0\}.$$

XVIII. Если выполняется неравенство

$$(f_k(0) - f_k(\delta_j^{(2)})) \cdot (1 - \beta) \geqslant \beta\Delta_j,$$

то положить $\rho_k = \delta_j^{(2)}$ и прекратить вычисления; иначе перейти к шагу XIX.

XIX. Вычислить γ_j — середину отрезка $[\delta_j^{(1)}, \delta_j^{(3)}]$

$$\gamma_j = (\delta_j^{(1)} + \delta_j^{(3)})/2.$$

XX. Вычислить точку $\delta_j^{(4)}$, симметричную $\delta_j^{(2)}$ относительно точки γ_j.

XXI. Если $f_k(\delta_j^{(4)}) < f_k(\delta_j^{(2)})$, то положить

$$\delta_{j+1}^{(1)} = \delta_j^{(2)};\quad \delta_{j+1}^{(2)} = \delta_j^{(4)};\quad \delta_{j+1}^{(3)} = \delta_j^{(3)}$$

и перейти к шагу XXII; иначе положить

$$\delta_{j+1}^{(1)} = \delta_j^{(1)};\quad \delta_{j+1}^{(2)} = \delta_j^{(2)};\quad \delta_{j+1}^{(3)} = \delta_j^{(4)}$$

и перейти к шагу XXII.

XXII. Положить $j = j + 1$ и перейти к шагу XVI.

Теорема 2. *Если выполнены условия теоремы 1 и функции* $f_k(t)$, $k = 0, 1, \ldots$, *выпуклы на множестве* $[0, \infty)$, *то любая предельная точка бесконечной последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 2, является стационарной точкой функции* $\max_{i \in \mathcal{I}} \varphi_i(x)$.

Алгоритм 3 применяется для нахождения ε-стационарных точек функции $\max_{i \in \mathcal{I}} \varphi_i(x)$. Если x^* — ε-стационарная точка и $\varphi_i(x)$, $i \in \mathcal{K}_\varepsilon(x^*)$ — выпуклые функции, то $\max_{i \in \mathcal{I}} \varphi_i(x^*)$ — приближенное значение для минимума функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ с абсолютной погрешностью, не превышающей ε.

Алгоритм 3′ основан на применении алгоритма 3 и дает возможность вычислять стационарные точки функции $\max_{i \in \mathcal{I}} \varphi_i(x)$.

Точка $x^* \in \boldsymbol{R}^n$ называется ε-стационарной точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$, если

$$\min_{\|g\|=1} \max_{i \in \mathcal{K}_\varepsilon(x^*)} \left(\frac{\partial \varphi_i(x^*)}{\partial x}, g \right) \geqslant 0$$

(здесь $g \in \boldsymbol{R}^n$, $\mathcal{K}_\varepsilon(x)$ определяется по (3.10)).

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$, константу $\varepsilon > 0$.

II. Положить $k = 0$.

Основной цикл. III. Найти множество индексов

$$\mathcal{K}_\varepsilon(x^k) \triangleq \{ i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon,\ i \in \mathcal{I} \}.$$

IV. Найти многогранник $L_\varepsilon(x^k)$, который является выпуклой оболочкой, натянутой на множество точек

$$\{ \nabla \varphi_i(x^k),\quad i \in \mathcal{K}_\varepsilon(x^k) \}.$$

V. Используя алгоритм 1′, определить, принадлежит ли начало координат многограннику $L_\varepsilon(x^k)$. Если начало координат принадлежит $L_\varepsilon(x^k)$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VI.

VI. Используя алгоритм 1″, найти точку z_ε, которая является ближайшей к началу координат точкой многогранника $L_\varepsilon(x^k)$.

VII. Вычислить вектор $h^k(\varepsilon)$ — направление ε-наискорейшего спуска функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ в точке x^k

$$h^k(\varepsilon) = -(1/\|z_\varepsilon\|) z_\varepsilon.$$

VIII. Вычислить шаговый множитель ρ_k из условия

$$\max_{i \in \mathcal{I}} \varphi_i(x^k + \rho_k h^k(\varepsilon)) = \min_{\rho \geqslant 0} \max_{i \in \mathcal{I}} \varphi_i(x^k + \rho h^k(\varepsilon)).$$

IX. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k(\varepsilon).$$

X. Положить $k = k + 1$ и перейти к шагу III.

Теорема 3. *Если выполнены условия теоремы 1, то любая предельная точка бесконечной последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 3, является ε-стационарной точкой функции* $\max_{i \in \mathcal{I}} \varphi_i(x)$.

Замечание 3. Вычисление шагового множителя ρ_k на шаге VIII алгоритма 3 из условия минимума практически неосуществимо. В [127] рекомендуется использовать алгоритм 2′ для вычисления за конечное число итераций шагового множителя ρ_k, удовлетворяющего условию (3.11). В случае, когда $f_k(t)$, $k = 0, 1, \ldots,$— выпуклые на $[0, \infty)$ функции, теорема 3 остается в силе.

Алгоритм 3 может быть также использован для нахождения стационарных точек функции $\max_{i \in \mathcal{I}} \varphi_i(x)$, только на шаге VI алгоритма 3 необходимо дополнительно вычислять значение

$$\psi_\varepsilon(x^k) = -\|z_\varepsilon\|.$$

Алгоритм 3′

Начало. I. Выбрать произвольное начальное приближение $x^{0,0} \in R^n$, константы $\varepsilon_0 > 0$, $\rho_0 > 0$.

II. Положить $l = 0$, $k_0 = 0$.

Основной цикл. III. Положить $x^0 = x^{l,k_l}$, $\varepsilon = \varepsilon_l$, $\rho = \rho_l$.

IV. Используя алгоритм 3, вычислить точку x^{k_l}, такую, что

$$\psi_\varepsilon(x^{k_l}) \geqslant -\rho$$

(такая точка получается за конечное число итераций алгоритма 3).

V. Положить $x^{l+1,k_{l+1}} = x^{k_l}$, $\varepsilon_{l+1} = \varepsilon_l/2$, $\rho_{l+1} = \rho_l/2$ и перейти к шагу III.

Теорема 3′. *Если выполнено предположение 0 и* $x^{0,0}$ *таково, что множество*

$$\{x \mid \max_{i \in \mathcal{I}} \varphi_i(x) \leqslant \max_{i \in \mathcal{I}} \varphi_i(x^{0,0}), \quad x \in R^n\}$$

ограничено, то любая предельная точка бесконечной последовательности $\{x^{l,k_l}\}_{l=0}^{\infty}$, *порожденной алгоритмом 3′, является стационарной точкой функции* $\max_{i \in \mathcal{I}} \varphi_i(x)$.

4. Модификация второго метода последовательных приближений

Предположение 4. Функции φ_i, $i \in \mathcal{I}$ — дважды непрерывно дифференцируемы на всем R^n.

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, константы $\alpha > 0$ и $\varepsilon > 0$.

II. Положить $k = 0$.

III. Найти множество

$$X_0 \triangleq \{x \mid \max_{i \in \mathcal{I}} \varphi_i(x) \leqslant \max_{i \in \mathcal{I}} \varphi_i(x^0), \quad x \in \boldsymbol{R}^n\}.$$

IV. Вычислить константу

$$\beta_1 = \max_{x \in X_0} \max_{i \in \mathcal{I}} \| \nabla \varphi_i(x) \|.$$

V. Найти множество

$$Y_0 = \{y \mid \|y\| \leqslant \alpha\beta_1, \ y \in \boldsymbol{R}^n\}.$$

VI. Вычислить константу

$$\beta_2 = \max_{x \in X_0, y \in Y_0} \max_{i \in \mathcal{I}} \left\| \frac{\partial^2 \varphi_i(x+y)}{\partial x^2} \right\|,$$

где

$$\left\| \frac{\partial^2 \varphi_i(x+y)}{\partial x^2} \right\| = \sqrt{\sum_{j_1, j_2 = 1}^{n} \left(\frac{\partial^2 \varphi_i(x+y)}{\partial x_{j_1} \partial x_{j_2}} \right)^2}.$$

VII. Если $\beta_2 = 0$, то вычислить константу $\bar{\alpha} = \min\{\alpha, \varepsilon/(2\beta_1^2)\}$ и перейти к шагу IX; иначе перейти к шагу VIII.

VIII. Вычислить константу

$$\alpha = \min\left\{\alpha, \ 2/\beta_2, \ \frac{-\beta_1 + \sqrt{\beta_1^2 + \beta_2 \varepsilon}}{\beta_1 \beta_2}\right\}.$$

IX. Выбрать произвольное $\alpha_0 \in (0, \bar{\alpha})$.

Основной цикл. X. Найти множество индексов

$$\mathcal{K}_\varepsilon(x^k) \triangleq \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \quad i \in \mathcal{I}\}.$$

XI. Используя алгоритм 1′, определить, принадлежит ли начало координат многограннику $L_\varepsilon(x^k)$, который является выпуклой оболочкой, натянутой на множество точек $\{\nabla \varphi_i(x^k), \ i \in \mathcal{K}_\varepsilon(x^k)\}$. Если начало координат принадлежит $L_\varepsilon(x^k)$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу XII.

XII. Используя алгоритм 1″, найти точку z_ε, которая является ближайшей к началу координат точкой многогранника $L_\varepsilon(x^k)$.

XIII. Вычислить $\rho_k = \| z_\varepsilon \|$.

XIV. Вычислить вектор

$$h^k(\varepsilon) = -(1/\|z_\varepsilon\|) z_\varepsilon.$$

XV. Вычислить следующее приближение

$$x^{k+1} = x^k + \alpha_0 \rho_k h^k(\varepsilon).$$

XVI. Положить $k = k + 1$ и перейти к шагу X.

Теорема 4. *Если выполнено предположение 4 и начальное приближение x^0 таково, что множество X_0 ограничено, то любая предельная точка бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 4, является ε-стационарной точкой функции* $\max_{i \in \mathcal{I}} \varphi_i(x)$.

5. Третий метод последовательных приближений, использующий D-функцию

Третий метод последовательных приближений, использующий D-функцию, весьма эффективный при малых n и небольшом количестве элементов множества $\mathcal{I}$.

Алгоритм 5

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Выбрать константу $\bar{\varepsilon} > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Найти множество индексов

$$\mathcal{K}_0(x^k) \triangleq \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) = \varphi_i(x^k), \quad i \in \mathcal{I}\}.$$

V. Используя алгоритм 1′, определить, принадлежит ли начало координат многограннику $L_0(x^k)$, который является выпуклой оболочкой, натянутой на множество точек $\{\nabla\varphi_i(x^k),\ i \in \mathcal{K}_0(x^k)\}$. Если начало координат принадлежит $L_0(x^k)$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VI.

VI. Найти множество чисел $\{\beta_j,\ j \in [0 : m]\}$, удовлетворяющее условиям: (i) — $\beta_0 > \beta_1 > \ldots > \beta_m$; (ii) — для каждого $i \in \mathcal{I}$ существует индекс $j \in [0 : m]$ такой, что $\varphi_i(x^k) = \beta_j$; (iii) — для каждого $j \in [0 : m]$ существует по крайней мере один индекс $i \in \mathcal{I}$ такой, что $\beta_j = \varphi_i(x^k)$.

VII. Положить $s = 0$, $\alpha_0 = 0$.

VIII. Положить $\varepsilon = \alpha_s$.

IX. Найти множество индексов

$$\mathcal{K}_\varepsilon(x^k) \triangleq \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \quad i \in \mathcal{I}\}.$$

X. Используя алгоритм 1″, найти точку z_ε — ближайшую к началу координат точку многогранника $L_\varepsilon(x^k)$, который является выпуклой оболочкой, натянутой на множество точек $\{\nabla\varphi_i(x^k),\ i \in \mathcal{K}_\varepsilon(x^k)\}$.

XI. Вычислить вектор $h^k(\varepsilon_s)$ — направление ε_s-наискорейшего спуска функции максимума в точке x^k

$$h^k(\varepsilon_s) = -(1/\|z_\varepsilon\|)\, z_\varepsilon.$$

XII. Вычислить значение

$$\psi_{\alpha_s}(x^k) = -\|z_\varepsilon\|.$$

XIII. Вычислить значение

$$\alpha_{s+1} \doteq \max_{i \in \mathcal{I}} \varphi_i(x^k) - \beta_{s+1}.$$

XIV. Если $\alpha_{s+1} < \bar{\varepsilon}$, то перейти к шагу XV; иначе перейти к шагу XVI;

XV. Если $s + 1 < m$, то положить $s = s + 1$ и перейти к шагу VIII; иначе перейти к шагу XVI.

XVI. Положить $\alpha_{s+1} = \bar{\varepsilon}$.

XVII. Найти индекс $s_k \in [0 : s]$ такой, что

$$\alpha_{s_k+1}\psi_{\alpha_{s_k}}(x^k) = \min \{\alpha_1\psi_{\alpha_0}(x^k),$$

$$\alpha_2\psi_{\alpha_1}(x^k), \ \ldots, \ \alpha_s\psi_{\alpha_{s-1}}(x^k), \ \alpha_{s+1}\psi_{\alpha_s}(x^k)\}.$$

XVIII. Положить $D(x^k) = \alpha_{s_k+1}\psi_{\alpha_{s_k}}$.

XIX. Положить $h^k = h^k(\varepsilon_{s_k})$.

XX. Найти шаговый множитель ρ_k из условия

$$\max_{i \in \mathcal{I}} \varphi_i(x^k + \rho_k h^k) = \min_{\rho \in [0, \infty)} \max_{i \in \mathcal{I}} \varphi_i(x^k + \rho h^k).$$

XXI. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

XXII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 5. *Если выполнены предположения 0 и начальное приближение x^0 в алгоритме 5 таково, что множество*

$$X_0 \triangleq \{x \mid \max_{i \in \mathcal{I}} \varphi_i(x) \leqslant \max_{i \in \mathcal{I}} \varphi_i(x^0), \quad x \in \boldsymbol{R}^n\}$$

ограничено, то любая предельная точка x^* бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 5, является стационарной точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$, причем $D(x^*) = 0$.

Библиографические указания. Пункты 1, 2 написаны на основании работ [117, 118, 127, 128, 254]. Пункты 3, 4 написаны на основании работ [117, 118, 127, 128, 36]. Пункт 5 основан на работах [119, 120, 122, 127, 128].

3.13. Сеточный метод последовательных приближений решения непрерывных минимаксных задач

Задача 1. Найти $\arg \min_{x \in R^n} \max_{y \in Y} \varphi(x, y)$ для заданной функции $\varphi : \boldsymbol{R}^n \times \boldsymbol{R}^m \to \boldsymbol{R}^1$ и заданного ограниченного замкнутого множества $Y \subset \boldsymbol{R}^m$.

Предположение 1. Функция $\varphi(x, y)$ непрерывна вместе с $\nabla_x\varphi(x, y)$ по совокупности переменных в $\boldsymbol{R}^n \times Y$.

Алгоритм 1

Начало. I. Выбрать начальное приближение $\bar{x}^0 \in \boldsymbol{R}^n$ и сетку

$$Y_{N_0} = \{y^i \mid y^i \in Y, \quad i \in [0:N_0]\},$$

удовлетворяющие условиям теоремы 1.

II. Выбрать произвольную константу $\alpha_0 > 0$.

III. Положить $s = 0$.

Основной цикл. IV. Положить $\alpha = \alpha_s$, $\mathcal{I} = [0:N_s]$.

V. Определить функции $\varphi_i : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $i \in \mathcal{I}$ по следующим правилам $\varphi_i(x) \triangleq \varphi(x, y^i)$, $i \in \mathcal{I}$.

VI. Если $s = 0$, то перейти к шагу VIII; иначе перейти к шагу VII.

VII. Если выполняется неравенство

$$\max_{i \in \mathcal{I}} \varphi_i(x^k) \leqslant \max_{i \in \mathcal{I}} \varphi_i(\bar{x}^0),$$

то положить $\bar{x}^s = x^k$ и перейти к шагу VIII; иначе положить $\bar{x}^s = \bar{x}^0$ и перейти к шагу VIII.

VIII. Используя алгоритм 12.5 (см. § 3.12, алгоритм 5) с начальным приближением $x^0 = \bar{x}^s$, за конечное число итераций вычислить первую точку x^k, для которой D-функция удовлетворяет условию

$$-D(x^k) \leqslant \alpha.$$

IX. Положить $N_{s+1} = 2N_s$, $\alpha_{s+1} = \alpha_s/2$.

X. Построить сетку

$$Y_{N_{s+1}} = \{y^i \mid y^i \in Y, \quad i \in [0:N_{s+1}]\},$$

удовлетворяющую условиям теоремы 1.

XI. Положить $s = s + 1$ и перейти к шагу IV.

Теорема 1. *Пусть выполнено предположение 1 и (**i**) — множество Y ограничено и замкнуто; (**ii**) — последовательность сеток $\{Y_{N_s}\}_{s=0}^{\infty}$ всюду плотная на множестве Y, причем $Y_{N_s} \subset Y_{N_{s+1}}$, $s = 0, 1, \ldots$; (**iii**) — начальное приближение $\bar{x}^0 \in \boldsymbol{R}^n$ и сетка Y_{N_0} таковы, что множество*

$$X_0 \triangleq \{x \mid \max_{i \in [0:N_0]} \varphi_i(x) \leqslant \max_{y \in Y} \varphi(\bar{x}^0, y), \quad x \in \boldsymbol{R}^n\},$$

ограничено, тогда любая предельная точка последовательности $\{\bar{x}^s\}_{s=0}^{\infty}$, порожденной алгоритмом 1, является стационарной точкой функции $\max_{y \in Y} \varphi(x, y)$ на $\boldsymbol{R}^n$.

Библиографические указания. При написании параграфа использовались работы [127, 120].

3.14. Методы Эрроу — Гурвица решения непрерывных минимаксных задач

1. Детерминированный метод Эрроу — Гурвица

Задача 1. Найти $\arg \min_{x \in R^n} \max_{y \in R^m} \varphi(x, y)$ для заданной функции $\varphi : R^n \times R^m \to R^1$.

Предположения 1. (*i*) — функция $\varphi(x, y)$ непрерывно дифференцируема по x и y; (*ii*) — функция $\varphi(x, y)$ выпукла по x при любом y и вогнута по y при любом x.

В методе Эрроу — Гурвица на k-й итерации вычисляют $(k+1)$-е приближение (x^{k+1}, y^{k+1}) в направлении антиградиента по x и градиента по y функции $\varphi(x, y)$, вычисленных в точке (x^k, y^k). Шаговые множители по переменным x и y удовлетворяют классическим условиям и могут быть различными.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, $y^0 \in R^m$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить векторы $\nabla_x \varphi(x^k, y^k)$ и $\nabla_y \varphi(x^k, y^k)$ — градиенты функции $\varphi(x, y)$, соответственно, по переменным x и y, вычисленные в точке (x^k, y^k).

IV. Вычислить шаговые множители ρ_k и ρ_k', удовлетворяющие условиям теоремы 1.

V. Вычислить следующие приближения:

$$x^{k+1} = x^k - \rho_k \nabla_x \varphi(x^k, y^k);$$

$$y^{k+1} = y^k + \rho_k' \nabla_y \varphi(x^k, y^k).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются предположения 1 и пусть (i) — у функции $\varphi(x, y)$ существует по крайней мере одна седловая точка; (ii) — шаговые множители ρ_k и ρ_k' алгоритма 1 удовлетворяют условиям*

$$\rho_k > 0 \text{ и } \rho_k' > 0 \text{ при } k = 0, 1, \ldots, \sum_{k=0}^{\infty} \rho_k = \infty;$$

$$\rho_k' \to 0, \ \rho_k / \rho_k' \to 0 \text{ и } \rho_{k+1}/\rho_k \to 1 \text{ при } k \to \infty.$$

Тогда, если последовательности $\{x^k\}_{k=0}^{\infty}$ и $\{y^k\}_{k=0}^{\infty}$, порожденные алгоритмом 1, ограничены, то любая предельная точка $\bar{x}$ последовательности $\{x^k\}_{k=0}^{\infty}$ является первой компонентой $\bar{x}$ некоторой седловой точки $(\bar{x}, \bar{y})$ функции $\varphi(x, y)$.

Замечание 1. При сделанных в теореме предположениях относительно последовательности $\{y^k\}_{k=0}^{\infty}$ можно утверждать лишь ее

сходимость к множеству

$$\tilde{Y} = \{\tilde{y} \mid \varphi(x^*, \tilde{y}) = \max_{y \in R^m} \varphi(x^*, y), \ \tilde{y} \in R^m\},$$

где x^* — вектор, пробегающий множество первых компонент седловых точек функции $\varphi(x, y)$.

В некоторых случаях $\tilde{Y}$ совпадает с множеством вторых компонент седловых точек функции $\varphi(x, y)$, однако в общем случае эти множества различны (примером этого служит функция $\varphi(x, y) = xy$).

2. Стохастический метод Эрроу — Гурвица

Задача 2. Найти $\arg \min_{x \in R^n} \max_{y \in R^m} E_\omega \varphi(x, y, \omega)$ для заданной функции $\varphi : R^n \times R^m \times \Omega \to R^1$.

Предположения 2. (*i*) — функция $f_0(x, y) \triangleq E_\omega \varphi(x, y, \omega)$ непрерывно дифференцируема по x и y; (*ii*) — функция $f_0(x, y)$ выпукла по x при любом y и вогнута по y при любом x.

В стохастическом методе Эрроу — Гурвица на k-й итерации вычисляют $(k+1)$-е приближение в направлении статистических оценок антиградиента по x и градиента по y функции $f_0(x, y)$ в точке (x^k, y^k). Шаговые множители по переменным x и y могут быть различны и удовлетворяют классическим условиям.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$, $y^0 \in R^m$

II. Положить $k = 0$.

Основной цикл. III. Вычислить шаговые множители ρ_k и ρ_k', удовлетворяющие условиям теоремы 2.

IV. Вычислить реализации ξ^k, ζ^k случайных векторов $\tilde{\xi}^k$, $\tilde{\zeta}^k$, удовлетворяющих условиям

$$E(\tilde{\xi}^k/\Phi_k) = \nabla_x f_0(x^k, y^k), \quad E(\|\tilde{\xi}^k\|^2/\Phi_k) \leqslant \alpha;$$

$$E(\tilde{\zeta}^k/\Phi_k) = \nabla_y f_0(x^k, y^k), \quad E(\|\tilde{\zeta}^k\|^2/\Phi_k) \leqslant \alpha,$$

где Φ_k — σ-алгебра, порождаемая случайными величинами x^0, y^0, x^1, y^1, ..., x^k, y^k; $\alpha < \infty$.

V. Вычислить следующие приближения:

$$x^{k+1} = x^k - \rho_k \xi^k;$$

$$y^{k+1} = y^k + \rho_k' \zeta^k.$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполняются предположения 2 и пусть шаговые множители* ρ_k *и* ρ'_k *удовлетворяют условиям*

$$\rho_k > 0, \quad \rho'_k > 0 \text{ при } k = 0, 1, \ldots;$$

$$\sum_{k=0}^{\infty} (\rho'_k)^2 < \infty, \quad \rho_k/\rho'_k \to 0 \text{ при } k \to \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

Тогда, если последовательности $\{x^k\}_{k=0}^{\infty}$ и $\{y^k\}_{k=0}^{\infty}$, *порождаемые алгоритмом 2, ограничены почти наверное, то любая предельная точка последовательности* $\{x^k\}_{k=0}^{\infty}$ *минимизирует функцию* $\max_{y \in R^m} f_0(x, y)$, *т. е. является решением задачи 2.*

Библиографические указания. Параграф написан на основании работ [272, 314].

3.15. Методы экстремального базиса для решения непрерывных минимаксных задач

З а д а ч а 0. Найти $\arg \min_{x \in R^n} \max_{y \in Y} \varphi(x, y)$ для заданной функции $\varphi : R^n \times R^m \to R^1$ и заданного компактного множества Y в R^m.

Предположения 0. (*i*) — функция $\varphi(x, y)$ непрерывна по x и y на $R^n \times Y$; (*ii*) — функция $\varphi(x, y)$ выпукла по x в R^n при любом $y \in Y$; (*iii*) — функция $\varphi(x, y)$ дважды непрерывно дифференцируема по x на $R^n \times Y$; (*iv*) — Y — компактное множество.

В методах экстремального базиса на k-й итерации вычисляют следующее $(k+1)$-е приближение x^{k+1} как решение (точное или приближенное) некоторой вспомогательной задачи безусловной минимизации. Для этого на каждой итерации строят базис из $n+2$ точек, принадлежащих множеству Y, в котором при переходе от одной итерации к другой меняется лишь одна точка. Вводимую в базис новую точку находят как решение (точное или приближенное) некоторой вспомогательной задачи условной максимизации. Для отыскания выводимой из базиса «лишней» точки в общем случае требуется иметь процедуру проверки на принадлежность этой точки к выпуклой оболочке заданных $n+1$ точек.

1. Принципиальный алгоритм

Алгоритм 1

Н а ч а л о. I. Выбрать произвольную точку $x^0 \in R^n$ и произвольный набор (базис) G_0, состоящий из $n+2$ точек множества Y

$$G_0 = \{y^{0,1}, y^{0,2}, \ldots, y^{0,n+2}\}, \quad y^{0,i} \in Y \quad \forall i \in [1 : n+2].$$

II. Положить $k = 0$.

О с н о в н о й ц и к л. III. Определить функцию

$$\psi_k(x) = \max_{i \in [1:n+2]} \varphi(x, y^{k,i}). \tag{3.13}$$

IV. Найти точку x^{k+1}, удовлетворяющую условию

$$\psi_k(x^{k+1}) = \min_{x \in R^n} \psi_k(x). \tag{3.14}$$

V. Найти точку $\tilde{y}^k \in Y$, удовлетворяющую условию

$$\varphi(x^{k+1}, \tilde{y}^k) = \max_{y \in Y} \varphi(x^{k+1}, y). \tag{3.15}$$

VI. Если выполняется равенство

$$\varphi(x^{k+1}, \tilde{y}^k) = \psi_k(x^{k+1}), \tag{3.16}$$

то прекратить вычисления (в этом случае x^{k+1} является решением задачи 0); иначе (если $\varphi(x^{k+1}, \tilde{y}^k) > \psi_k(x^{k+1})$) перейти к шагу VII.

VII. Вычислить множество индексов

$$J_k = \{i \mid \varphi(x^{k+1}, y^{k,i}) = \psi_k(x^{k+1}), \quad i \in [1 : n+2]\}.$$

Если множество J_k содержит все элементы множества $[1 : n+2]$, то перейти к шагу VIII; иначе обозначить через i_k любой индекс из множества $[1 : n+2]$, которого нет в множестве J_k, и перейти к шагу IX.

VIII. Найти индекс $i_k \in [1 : n+2]$ такой, что $0 \in \hat{L}_k$, где множество $\hat{L}_k$ является выпуклой оболочкой множества

$$\hat{H}_k \triangleq \{\nabla_x \varphi(x^{k+1}, y^{k,i}) \mid i \in (J_k \setminus \{i_k\})\}.$$

(Отметим, что по необходимому условию минимума функции $\psi_k(x)$ точка x^{k+1} удовлетворяет условию $0 \in L_k$, где множество L_k является выпуклой оболочкой множества

$$H_k \triangleq \{\nabla_x \varphi(x^{k+1}, y^{k,i}) \mid i \in J_k\}.$$

Поэтому начало координат (точку 0) можно представить в виде выпуклой комбинации не более чем $n+1$ векторов

$$\nabla_x \varphi(x^{k+1}, y^{k,i}), \ i \in (J_k \setminus \{i_k\})).$$

IX. Построить новый базис $G_{k+1} = \{y^{k+1,1}, \ldots, y^{k+1,n+2}\}$ по правилу

$$y^{k+1,i} = \begin{cases} y^{k,i}, & \text{если } i \neq i_k; \\ \tilde{y}^k, & \text{если } i = i_k, \end{cases} \quad i = 1, \ldots, n+2. \tag{3.17}$$

X. Положить $k = k+1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются предположения 0 и пусть существуют числа $\alpha_1 > 0$ и $\alpha_2 < \infty$ такие, что*

$$(\nabla^2_{xx}\varphi(x, y) \cdot z, z) \geqslant \alpha_1 \|z\|^2, \quad \forall x \in R^n, \ y \in Y, \ z \in R^n;$$

$$\|\nabla_x \varphi(x, y)\| \leqslant \alpha_2, \quad \forall x \in R^n, \ y \in Y.$$

Тогда бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к точке минимума функции $\max_{\in Y} \varphi(x, y)$.

2. ε-метод

В общем случае ε-метод практически не реализуем из-за бесконечности процедуры безусловной минимизации.

Алгоритм 2

Начало. I, II (шаги I и II такие, как в алгоритме 1).

III. Выбрать последовательность $\{\varepsilon_k\}_{k=0}^{\infty}$, удовлетворяющую условиям

$$\varepsilon_k \to +0 \text{ при } k \to \infty, \quad \sum_{k=0}^{\infty} \varepsilon_k < \infty.$$

Основной цикл. IV. Определить функцию $\psi_k(x)$ по (3.13).

V. Вычислить точку x^{k+1} такую, что $0 \in L_{k,\varepsilon_k}$, где L_{k,ε_k} является выпуклой оболочкой множества точек

$$H_{k,\varepsilon_k} \triangleq \{\nabla_x \varphi(x^{k+1}, y^{k,i}) \mid i \in J_{k,\varepsilon_k}\},$$

здесь

$$J_{k,\varepsilon_k} = \{i \mid \varphi(x^{k+1}, y^{k,i}) \geqslant \psi_k(x^{k+1}) - \varepsilon_k, \ i \in [1 : n+2]\}.$$

VI. Вычислить точку $\tilde{y}^k \in Y$ по (3.15).

VII. Если выполняется равенство (3.16), то прекратить вычисления (в этом случае x^{k+1} является решением задачи 0); иначе перейти к шагу VIII.

VIII. Если множество J_{k,ε_k} содержит все элементы множества $[1 : n+2]$, то перейти к шагу IX; иначе обозначить через i_k любой индекс множества $[1 : n+2]$, которого нет в множестве J_{k,ε_k}, и перейти к шагу X.

IX. Найти индекс $i_k \in [1 : n+2]$ такой, что $0 \in \hat{L}_{k,\varepsilon_k}$, где множество $\hat{L}_{k,\varepsilon_k}$ является выпуклой оболочкой множества

$$\hat{H}_{k,\varepsilon_k} \triangleq \{\nabla_x \varphi(x^{k+1}, y^{k,i}) \mid i \in (J_{k,\varepsilon_k} \setminus \{i_k\})\}.$$

X. Построить новый базис $G_{k+1} = \{y^{k+1,1}, \ldots, y^{k+1,n+2}\}$ по (3.17).

XI. Положить $k = k+1$ и перейти к шагу IV.

Для алгоритма 2 имеет место теорема, аналогичная теореме 1.

Замечание 2. На шаге VI алгоритма 2 точку $\tilde{y}^k \in Y$ можно вычислять по правилу

$$\max_{y \in Y} \varphi(x^{k+1}, y) - \varphi(x^{k+1}, \tilde{y}^k) \leqslant \mu_{k+1}, \tag{3.18}$$

где $\mu_{k+1} \to +0$ при $k \to \infty$, что требует приближенного решения задачи максимизации по $y \in Y$ функции $\varphi(x^{k+1}, y)$.

3. ρ-метод

Приводимый ρ-метод реально осуществимый, так как требует лишь приближенного решения промежуточных задач оптимизации.

Алгоритм 3

Начало. I, II (шаги I и II такие, как в алгоритме 1).

III. Выбрать последовательность $\{\rho_k\}_{k=0}^{\infty}$, удовлетворяющую условию

$$\rho_k \to +0 \text{ при } k \to \infty, \quad \sum_{k=0}^{\infty} \rho_k < \infty.$$

Основной цикл. IV. Определить функцию $\psi_k(x)$ по (3.13).

V. Вычислить точку x^{k+1} такую, что

$$\min_{z \in L_k} \|z\| \leqslant \rho_k,$$

где L_k является выпуклой оболочкой множества

$$H_k \triangleq \{\nabla_x \varphi(x^{k+1}, y^{k,i}), \ i \in J_k\};$$

здесь

$$J_k = \{i \mid \varphi(x^{k+1}, y^{k,i}) = \psi_k(x^{k+1}), \ i \in [1 : n+2]\}.$$

(Для вычисления x^{k+1} требуется решать приближенно задачу минимизации функции $\psi_k(x)$ в R^n).

VI. Вычислить точку $\tilde{y}^k \in Y$ по формуле (3.15) (как и в замечании 2 точку $\tilde{y}^k$ можно также вычислять в соответствии с неравенством (3.18), т. е. для отыскания $\tilde{y}^k$ достаточно решать приближенно задачу максимизации по $y \in Y$ функции $\varphi(x^{k+1}, y)$).

VII. Если $J_k = [1 : n+2]$, то перейти к шагу VIII; иначе в качестве i_k выбрать любой индекс из множества $[1 : n+2] \setminus J_k$ и перейти к шагу IX.

VIII. Найти индекс $i_k \in [1 : n+2]$ такой, что множества L_k и $\hat{L}_k$ совпадают, где L_k определяется на шаге V, а $\hat{L}_k$ является выпуклой оболочкой множества

$$\hat{H}_k \triangleq \{\nabla_x \varphi(x^{k+1}, y^{k,i}) \mid i \in (J_k \setminus \{i_k\})\}.$$

IX. Построить новый базис $G_{k+1} = \{y^{k+1,1}, \ldots, y^{k+1,n+2}\}$ по формуле (3.17).

X. Положить $k = k + 1$ и перейти к шагу IV.

Для алгоритма 3 имеет место теорема, аналогичная теореме 1.

4. Комбинированный ε,ρ-метод

Приводимый здесь алгоритм является комбинацией методов, приведенных в пунктах 2, 3, и является практически реализуемым

Алгоритм 4

Начало. I, II (шаги I и II такие, как в алгоритме 1).

III. Выбрать последовательности $\{\varepsilon_k\}_{k=0}^{\infty}$, $\{\rho_k\}_{k=0}^{\infty}$, удовлетворяющие условиям

$$\varepsilon_k \to +0, \; \rho_k \to +0 \text{ при } k \to \infty;$$

$$\sum_{k=0}^{\infty} \varepsilon_k < \infty, \quad \sum_{k=0}^{\infty} \rho_k < \infty.$$

Основной цикл. IV. Определить функцию $\psi_k(x)$ по формуле (3.13).

V. Вычислить точку x^{k+1} такую, что

$$\min_{z \in L_{k,\varepsilon_k}} \|z\| \leqslant \rho_k,$$

где L_{k,ε_k} определяется, как на шаге V алгоритма 2.

Для вычисления x^{k+1} необходимо решить приближенно задачу минимизации функции $\psi_k(x)$ в R^n.

VI. Вычислить точку $\tilde{y}^k$ по (3.15).

По аналогии с замечанием 2 для отыскания $\tilde{y}^k$ достаточно решить приближенно задачу максимизации по $y \in Y$ функции $\varphi(x^{k+1}, y)$.

VII. Если $J_{k,\varepsilon_k} = [1 : n+2]$, то перейти к шагу VIII; иначе в качестве i_k выбрать любой индекс из множества $[1 : n+2] \diagdown J_{k,\varepsilon_k}$ и перейти к шагу IX.

VIII. Найти индекс $i_k \in [1 : n+2]$ такой, что множества L_{k,ε_k} и $\hat{L}_{k,\varepsilon_k}$ совпадают, где L_{k,ε_k} определяется, как на шаге V алгоритма 2, а $\hat{L}_{k,\varepsilon_k}$ — как на шаге IX алгоритма 2.

IX. Построить новый базис $G_{k+1} = \{y^{k+1,1}, \ldots, y^{k+1,n+2}\}$ по (3.17).

X. Положить $k = k+1$ и перейти к шагу IV.

Для алгоритма 4 имеет место теорема, аналогичная теореме 1.

Замечание 4. Если в алгоритмах пунктов 2, 3, 4 точку $\tilde{y}^k$ вычислять в соответствии с неравенством (3.18), где $\mu_k \equiv \mu > 0$, то последовательности точек $\{x^k\}_{k=0}^{\infty}$, порождаемые этими алгоритмами, будут сходиться к μ-стационарной точке функции $\max_{y \in Y} \varphi(x, y)$ (определение μ-стационарной точки приведено в § 3.12).

Библиографические указания. При написании параграфа была использована работа [125]. Дополнительную информацию о методах экстремального базиса можно найти в работах [124, 125].

3.16. Обобщенный градиентный метод отыскания седловых точек

Задача 1. Найти седловую точку (x^*, y^*), удовлетворяющую соотношениям

$$\varphi(x^*, y^*) = \min_{x \in R^n} \max_{y \in R^m} \varphi(x, y) = \max_{y \in R^m} \min_{x \in R^n} \varphi(x, y),$$

где $\varphi : R^n \times R^m \to R^1$ — заданная функция.

Предположения 1. (*i*) — для каждого фиксированного вектора $y \in R^m$ функция $\varphi(x, y)$ выпукла и непрерывно дифференцируема по x; (*ii*) — для каждого фиксированного вектора $x \in R^n$ функция $\varphi(x, y)$ вогнута по y; (*iii*) — множество седловых точек $X^* \times Y^*$ задачи 1 непусто.

Приводимый ниже метод является вариантом метода Эрроу — Гурвица. На k-й итерации алгоритма движение к следующему приближению (x^{k+1}, y^{k+1}) по переменной x осуществляется в направлении антиградиента по x функции $\varphi(x, y^k)$, а по переменной y — в направлении обобщенного градиента по y функции $\varphi(x^k, y)$. Шаговые множители удовлетворяют классическим условиям. При определенных условиях алгоритм сходится к неустойчивому по Гольштейну [82] множеству седловых точек функции $\varphi(x, y)$ (определение устойчивости множества седловых точек функции $\varphi(x, y)$ приведено в § 6.9).

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $(x^0, y^0) \in R^n \times R^m$.

II. Положить $k = 0$.

III. Выбрать константу $\alpha < \infty$, удовлетворяющую условиям теоремы 1.

Основной цикл. IV. Вычислить n-мерный вектор $\nabla_x \varphi(x^k, y^k)$ — градиент функции $\varphi(x, y)$ по x в точке (x^k, y^k).

V. Вычислить m-мерный вектор $\hat{\nabla}_y \varphi(x^k, y^k)$ — обобщенный градиент функции $\varphi(x, y)$ по y в точке (x^k, y^k).

VI. Вычислить шаговые множители ρ_k и ρ'_k, удовлетворяющие условиям теоремы 1.

VII. Вычислить следующие приближения:

$$x^{k+1} = \begin{cases} x^k - \rho_k \nabla_x \varphi(x^k, y^k), & \text{если } (x^k, y^k) \in S_x \times S_y, \\ x^0, & \text{если } (x^k, y^k) \notin S_x \times S_y; \end{cases}$$

$$y^{k+1} = \begin{cases} y^k + \rho'_k \hat{\nabla}_y \varphi(x^k, y^k), & \text{если } (x^k, y^k) \in S_x \times S_y, \\ y^0, & \text{если } (x^k, y^k) \notin S_x \times S_y, \end{cases}$$

где

$$S_x \triangleq \{x \mid \|x\| \leqslant \alpha,\ x \in R^n\},\quad S_y \triangleq \{y \mid \|y\| \leqslant \alpha,\ y \in R^m\}.$$

VIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. Пусть выполняются предположения 1 и существует константа $\alpha < \infty$ такая, что (iv) — при $x \notin S_x$ и для всех $y \in S_y$ выполняется неравенство $(\nabla_x \varphi(x, y), x) > 0$; (v) при $y \notin S_y$ и для всех $x \in S_x$ выполняется неравенство

$$\max_{\{\hat{\nabla}_y \varphi(x,y)\}} (\hat{\nabla}_y \varphi(x, y), y) < 0;$$

(*vi*)

$$\|\hat{\nabla}_y \varphi(x, y)\| \leqslant \beta_1 < \infty, \quad \forall x \in S_x, \quad y \in S_y.$$

Если шаговые множители ρ_k и ρ_k' в алгоритме 1 удовлетворяют условиям

$$\rho_k \to +0, \quad \rho_k' \to +0, \quad \rho_k'/\rho_k \to 0 \text{ при } k \to \infty;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k' = \infty$$

и начальное приближение (x^0, y^0) принадлежит множеству $S_x \times S_y$, то все предельные точки последовательности $\{y^k\}$, порожденной алгоритмом 1, принадлежат проекции на R^m множества седловых точек функции $\varphi(x, y)$, т. е. множеству Y^*.

Замечание 1. Пусть в дополнение к условиям теоремы 1 множество седловых точек $X^* \times Y^*$ функции $\varphi(x, y)$ устойчиво лишь по x (т. е. при $y^* \in Y^*$ минимум функции $\varphi(x, y^*)$ достигается лишь на точках множества X^*). Тогда алгоритм 1 сходится по переменным x и y к множеству седловых точек $X^* \times Y^*$.

Библиографические указания. Параграф написан на основании работы [273].

3.17. Квазиградиентные методы решения дискретных минимаксных задач стохастического программирования

1. Минимизация функции $E_\omega \max_{i \in \mathcal{I}} \varphi_i(x, \omega)$

Задача 1. Найти $\arg\min_{x \in R^n} E_\omega \max_{i \in \mathcal{I}} \varphi_i(x, \omega)$ для заданных функций $\varphi_i : R^n \times \Omega \to R^1$, $i \in \mathcal{I}$ и заданного множества индексов $\mathcal{I}$.

Предположения 1. (*i*) — функции φ_i, $i \in \mathcal{I}$ непрерывно дифференцируемы по x, причем градиенты по x функции φ_i, $i \in \mathcal{I}$ удовлетворяют при каждом ω локальному условию Липшица

$$\|\nabla_x \varphi_i(x, \omega) - \nabla_x \varphi_i(y, \omega)\| \leqslant \gamma_Z(\omega) \|x - y\|, \quad i \in \mathcal{I},$$

с интегрируемой константой Липшица $\gamma_Z(\omega)$, удовлетворяющей условию $E\gamma_Z(\omega) < \infty$ для x, y, принадлежащих замкнутому ограниченному множеству Z; (*ii*) — при каждом ω имеется возможность вычислять значения функций $\varphi_i(x, \omega)$, $i \in \mathcal{I}$ и их производных по x: $\nabla_x \varphi_i(x, \omega)$, $i \in \mathcal{I}$.

В алгоритме 1 на k-й итерации за вектор, определяющий направление движения к следующему приближению $x^{k+1}(\omega)$, выбирается $\nabla_x \varphi_{i_k}(x^k(\omega), \omega^k)$, где индекс $i_k \in \mathcal{I}$ удовлетворяет условию

$$\varphi_{i_k}(x^k, \omega^k) = \max_{i \in \mathcal{I}} \varphi_i(x^k, \omega^k).$$

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать константу $\delta > 0$ такую, чтобы выполнялось неравенство

$$\inf_{\|x\| \geq \delta} E \max_{i \in \mathcal{I}} \varphi_i(x, \omega) > E \max_{i \in \mathcal{I}} \varphi_i(x, \omega) \text{ при } \|x\| < \delta.$$

III. Положить $k = 0$.

Основной цикл. IV. Вычислить ω^k — реализацию случайного элемента ω.

V. Вычислить индекс i_k, удовлетворяющий условию

$$\varphi_{i_k}(x^k(\omega), \omega^k) = \max_{i \in \mathcal{I}} \varphi_i(x^k(\omega), \omega^k).$$

VI. Вычислить $\nabla_x \varphi_{i_k}(x^k(\omega), \omega^k)$.

VII. Найти шаговый множитель ρ_k, удовлетворяющий условиям теоремы 1.

VIII. Если $\|x^k(\omega)\| > \delta$, то положить $x^{k+1}(\omega) = x^k(\omega)$ и перейти к шагу X; иначе перейти к шагу IX.

IX. Вычислить

$$x^{k+1}(\omega) = x^k(\omega) - \rho_k \nabla_x \varphi_{i_k}(x^k(\omega), \omega^k).$$

X. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Если выполнены предположения 1 и (iii) —* $E \max_{i \in \mathcal{I}} \varphi_i \times (x, \omega) \to \infty$ *при* $\|x\| \to \infty$; *(iv) — функция* $E \min_{i \in \mathcal{I}} \varphi_i(x, \omega)$ *принимает на множестве решений* X^* *задачи 1 не более чем счетное число значений; (v) — последовательность шаговых множителей* $\{\rho_k\}_{k=0}^{\infty}$ *удовлетворяет условиям*

$$\rho_k > 0, \quad k = 0, 1, \ldots, \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty;$$

$$\rho_{k+1}/\rho_k \to 1 \quad \text{при} \quad k \to \infty,$$

то почти для всех ω *предельные точки последовательности* $\{x^k(\omega)\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, принадлежат множеству решений* X^* *задачи 1.*

2. Минимизация функции $\max\limits_{i \in \mathcal{I}} E\omega\varphi_i(x, \omega)$

Задача 2. Найти $\arg\min\limits_{x \in R^n} \max\limits_{i \in \mathcal{I}} E\varphi_i(x, \omega)$ для заданных функций $\varphi_i : R^n \times \Omega \to R^1$, $i \in \mathcal{I}$ и заданного множества индексов $\mathcal{I}$.

Предположения 2. Функции $\varphi_i(x, \omega)$, $i \in \mathcal{I}$ и распределение ω таковы, что выполняются следующие условия: (*i*) — существует константа $\delta < \infty$ такая, что $(\nabla f_i(x), x) > 0$ для $\|x\| > \delta$, где $f_i(x) \triangleq E\varphi_i(x, \omega)$; (*ii*) — градиенты функций $f_i(x)$, $i \in \mathcal{I}$, удовлетворяют локальному условию Липшица

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leqslant \gamma_Z \|x - y\|, \quad \gamma_Z < \infty$$

для всех x, y, принадлежащих произвольному замкнутому ограниченному множеству Z; (*iii*) — при каждом ω имеется возможность вычислять значение функций $\varphi_i(x, \omega)$, $i \in \mathcal{I}$ и их производных по x — $\nabla_x \varphi_i(x, \omega)$, $i \in \mathcal{I}$.

В алгоритме 2 на k-й итерации за вектор, определяющий направление движения к следующему приближению $x^{k+1}(\omega)$, выбирается $\nabla_x \varphi_{i_k}(x^k(\omega), \omega^k)$, где индекс $i_k \in \mathcal{I}$ удовлетворяет условию $z_{i_k}^k = \max\limits_{i \in \mathcal{I}} z_i^k$ (здесь при каждом $i \in \mathcal{I}$ вспомогательная последовательность чисел $\{z_i^k\}_{k=0}^{\infty}$ обладает свойством $z_i^k - f_i(x^k(\omega)) \to 0$ при $k \to \infty$).

Алгоритм 2

Начало. I. Выбрать константу $\delta > 0$, удовлетворяющую условию (*i*) предположения 2.

II. Выбрать произвольное приближение $x^0 \in R^n$, удовлетворяющее условию $\|x^0\| \leqslant \delta$.

III. Выбрать произвольные числа z_i^0, $i \in \mathcal{I}$.

IV. Положить $k = 0$.

Основной цикл. V. Вычислить индекс $i_k \in \mathcal{I}$ из условия

$$z_{i_k}^k(\omega) = \max_{i \in \mathcal{I}} z_i^k(\omega).$$

VI. Вычислить ω^k — реализацию случайного элемента ω.

VII. Вычислить $\nabla_x \varphi_{i_k}(x^k(\omega), \omega^k)$.

VIII. Найти шаговый множитель ρ_k и параметр σ_k, удовлетворяющие условиям теоремы 2.

IX. Вычислить следующее приближение

$$x^{k+1}(\omega) = \begin{cases} x^0, & \text{если } \|x^k(\omega)\| > 2\delta; \\ x^k(\omega) - \rho_k \nabla_x \varphi_{i_k}(x^k(\omega), \omega^k), & \text{если } \|x^k(\omega)\| \leqslant 2\delta. \end{cases}$$

X. Вычислить значения величин

$$z_i^{k+1}(\omega) = z_i^k(\omega) + \sigma_k(\varphi_i(x^k(\omega), \omega^k) - z_i^k(\omega)), \quad i \in \mathcal{I}.$$

XI. Положить $k = k + 1$ и перейти к шагу V.

Теорема 2. *Если выполнены предположения 2 и* (**iv**) — $E|\varphi_i \times (x, \omega)|^2 < \infty$, $E\|\nabla_x \varphi_i (x, \omega)\|^2 < \infty$, $i \in \mathcal{I}$; (*v*) — *последовательности шаговых множителей* $\{\rho_k\}_{k=0}^{\infty}$ *и параметров* $\{\sigma_k\}_{k=0}^{\infty}$ *удовлетворяют условиям*

$$\rho_k > 0, \quad k = 0, 1, \ldots, \quad 0 \leqslant \sigma_k \leqslant 1, \quad k = 0, 1, \ldots;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \sigma_k^2 < \infty;$$

$$\rho_k/\sigma_k \to 0 \ \text{ при } \ k \to \infty, \quad \rho_{k+1}/\rho_k \to 1 \ \text{ при } \ k \to \infty;$$

(**vi**) — *функция* $\max_{i \in \mathcal{I}} E\varphi_i (x, \omega)$ *принимает на множестве решений* X^* *задачи 2 не более чем счетное число значений, то почти для всех* ω *предельные точки последовательности* $\{x^k (\omega)\}_{k=0}^{\infty}$, *порожденной алгоритмом 2, принадлежат множеству решений* X^* *задачи 2.*

Библиографические указания. При написании параграфа использовались работы [158, 272].

Часть II

МЕТОДЫ УСЛОВНОЙ ОПТИМИЗАЦИИ

Глава 4

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Общая задача линейного программирования является задачей условной оптимизации, в которой целевая функция и функции ограничений линейны: найти $\arg\max\limits_{x} \sum\limits_{i=1}^{n} c_i x_i$ при ограничениях

$$\left.\begin{array}{l} \alpha_{11}x_1 + \alpha_{12}x_2 + \cdots + \alpha_{1n}x_n = a_1^0; \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ \alpha_{m_1 1}x_1 + \alpha_{m_1 2}x_2 + \cdots + \alpha_{m_1 n}x_n = a_{m_1}^0; \\ \alpha_{m_1+1,1}x_1 + \alpha_{m_1+1,2}x_2 + \cdots + \alpha_{m_1+1,n}x_n \leqslant a_{m_1+1}^0; \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \alpha_{m1}x_1 + \alpha_{m2}x_2 + \cdots + \alpha_{mn}x_n \leqslant a_m^0, \end{array}\right\} \quad (4.1)$$

$$m_1 \leqslant m;$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n_1; \quad n_1 \leqslant n, \quad (4.2)$$

для заданных $c = (c_1, c_2, \ldots, c_n)$; α_{ij}, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots$ $\ldots, n$; $a^0 = (a_1^0, a_2^0, \ldots, a_m^0)$. Ограничения (4.1), заданные в виде равенств и неравенств, называют общими; ограничения (4.2) — прямыми. Задача линейного программирования называется канонической, если в (4.1) $m_1 = m$ и в (4.2) $n_1 = n$, т. е. если общие ограничения состоят из равенств, а требование неотрицательности распространяется на все переменные x_j, $j = 1, \ldots, n$.

Если ввести дополнительные переменные $x_{n+1}, \ldots, x_{n+m-m_1}$ и сделать замену

$$\begin{array}{lll} x_{n_1+1} = x'_{n_1+1} - x''_{n_1+1}, & x'_{n_1+1} \geqslant 0, & x''_{n_1+1} \geqslant 0; \\ x_{n_1+2} = x'_{n_1+2} - x''_{n_1+2}, & x'_{n_1+2} \geqslant 0, & x''_{n_1+2} \geqslant 0; \\ \ldots\ldots\ldots\ldots & & \\ x_n = x'_n - x''_n, & x'_n \geqslant 0, & x''_n \geqslant 0, \end{array}$$

то общая задача линейного программирования сведется к эквивалентной канонической задаче:

$$\text{найти } \arg\max\,(c_1x_1 + \cdots + c_{n_1}x_{n_1} + c_{n_1+1}(x'_{n_1+1} - x''_{n_1+1}) + \cdots$$
$$\cdots + c_n(x'_n - x''_n))$$

при ограничениях

$$\alpha_{11}x_1 + \cdots + \alpha_{1n_1}x_{n_1} + \alpha_{1,n_1+1}(x'_{n_1+1} - x''_{n_1+1}) + \cdots$$
$$\cdots + \alpha_{1n}(x'_n - x''_n) = a_1^0;$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\alpha_{m_1 1}x_1 + \cdots + \alpha_{m_1 n_1}x_{n_1} + \alpha_{m_1,n_1+1}(x'_{n_1+1} - x''_{n_1+1}) + \cdots$$
$$\cdots + \alpha_{m_1 n}(x'_n - x''_n) = a_{m_1}^0;$$
$$\alpha_{m_1+1,1}x_1 + \cdots + \alpha_{m_1+1,n_1}x_{n_1} + \alpha_{m_1+1,n_1+1}(x'_{n_1+1} - x''_{n_1+1}) + \cdots$$
$$\cdots + \alpha_{m_1+1,n}(x'_n - x''_n) + x_{n+1} = a_{m_1+1}^0;$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\alpha_{m1}x_1 + \cdots + \alpha_{mn_1}x_{n_1} + \alpha_{m,n_1+1}(x'_{n_1+1} - x''_{n_1+1}) + \cdots$$
$$\cdots + \alpha_{mn}(x'_n - x''_n) + x_{n+m-m_1} = a_m^0.$$

Каноническая форма задачи линейного программирования является наиболее простой и удобной при построении вычислительных алгоритмов. Большинство приводимых в этой главе методов относится к канонической задаче линейного программирования. Те случаи, когда решение задачи с естественной формой записи сокращает трудоемкость численного анализа, будут рассмотрены вместе с соответствующими модификациями алгоритмов.

4.1. Симплекс-метод и его варианты

Задача 0. Найти $\arg\max_{x \in X}(c, x)$ для заданного вектора $c = (c_1, \ldots, c_n)$ и заданного множества

$$X \triangleq \{x \mid Ax = a^0,\ x \geqslant 0,\ x \in \boldsymbol{R}^n\},$$

здесь A — матрица размера $m \times n$:

$$A = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \ldots & \alpha_{1n} \\ \alpha_{21} & \alpha_{22} & \ldots & \alpha_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \alpha_{m1} & \alpha_{m2} & \ldots & \alpha_{mn} \end{pmatrix}, \quad a^0 = (a_1^0, a_2^0, \ldots, a_m^0).$$

Столбцы матрицы A обозначаются через $a^1, a^2, \ldots, a^n$.

Предположения 0. (*i*) — ранг матрицы A равен m; (*ii*) — $n > m$; (*iii*) — множество X непусто.

Определение 1. Допустимое решение x (т. е. вектор $x \in X$) называется опорным решением задачи 0, если система векторов a^j, соответствующая его положительным компонентам ($x_j > 0$), линейно-независима.

Определение 2. Базисом опорного решения x называют систему m линейно-независимых векторов, которая включает все векторы a^j, отвечающие положительным составляющим опорного решения x.

Определение 3. Опорное решение x называется невырожденным, если число его положительных компонент равно m (если оно меньше m, то опорное решение называется вырожденным).

Определение 4. Каноническая задача линейного программирования называется невырожденной, если все ее опорные решения невырождены.

Определение 5. Компоненты опорного решения, отвечающие векторам его базиса, называются базисными, а остальные — небазисными.

В дальнейшем базисные векторы будем обозначать через

$$a^{i_1},\ a^{i_2},\ \ldots,\ a^{i_m};$$

$$i_k \in [1:n],\quad k = 1,\ \ldots,\ m.$$

Базисные векторы a^{i_k} характеризуются номером i_k и позицией k, которую он занимает в базисе. Матрица B, составленная из векторов a^{i_1}, a^{i_2}, ..., a^{i_m}, называется базисной

$$B = (a^{i_1},\ \ldots,\ a^{i_m}).$$

В симплекс-методе решение задачи 0 начинается с известного опорного решения и его базиса. На каждой итерации алгоритма проводится проверка опорного решения на оптимальность. Если опорное решение не оптимально, то указывается способ, позволяющий по данному опорному построить другое опорное решение, более близкое к оптимальному. Через конечное число итераций либо находится решение задачи 0, либо устанавливается неограниченность целевой функции задачи 0 на множестве X.

1. Симплекс-метод решения невырожденной канонической задачи линейного программирования

Алгоритм 1

Начало. I. Найти опорное решение $x^1 = (x_1^1, \ldots, x_n^1)$ задачи 0, базис a^{i_1}, a^{i_2}, ..., a^{i_m} этого опорного решения и вычислить матрицу B^{-1}, обратную к исходной базисной матрице B, состоящей из столбцов a^{i_1}, a^{i_2}, ..., a^{i_m}.

Обычно на практике находят опорное решение, которому соответствует единичный базис $a^{i_1} = (1, 0, \ldots, 0)^T, \ldots, a^{i_m} = (0, 0, \ldots, 1)^T$. Для нахождения исходного базиса и опорного решения используют алгоритм 2.

II. Разложить по исходному базису a^{i_1}, a^{i_2}, ..., a^{i_m} векторы a^0, a^1, ..., a^n (т. е. найти числа z_{kj}, $k = 1, 2, \ldots, m$; $j = 0, 1, 2, \ldots, n$, такие, что

$$a^0 = z_{10}a^{i_1} + z_{20}a^{i_2} + \cdots + z_{m0}a^{i_m};$$

$$a^1 = z_{11}a^{i_1} + z_{21}a^{i_2} + \cdots + z_{m1}a^{i_m};$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$a^n = z_{1n}a^{i_1} + z_{2n}a^{i_2} + \cdots + z_{mn}a^{i_m})$$

по формулам

$$z_{10} = x^1_{i_1},\ z_{20} = x^1_{i_2},\ \ldots,\ z_{m0} = x^1_{i_m};$$

$$(z_{1j},\ z_{2j},\ \ldots,\ z_{mj})^T = B^{-1}(\alpha_{1j},\ \alpha_{2j},\ \ldots,\ \alpha_{mj})^T,\quad j = 1,\ \ldots,\ n.$$

Отметим, что заранее известно разложение по исходному базису базисных векторов a^{i_k}, $k = 1, \ldots, m$:

$$z_{li_k} = 0 \text{ при } k \neq l;\ z_{li_k} = 1 \text{ при } k = l,\ k = 1,\ 2,\ \ldots,\ m;$$
$$l = 1,\ 2,\ \ldots,\ m.$$

Основной цикл. III. Для каждого $j \in [1 : n]$ вычислить оценку

$$\Delta_j = \sum_{k=1}^{m} c_{i_k} z_{kj} - c_j. \tag{4.3}$$

IV. Если все $\Delta_j \geqslant 0$ ($j = 1, \ldots, n$), то положить $x^* = x^1$ и прекратить вычисления (в этом случае опорное решение оптимально и оптимум целевой функции равен $\sum_{k=1}^{m} c_{i_k} z_{k0}$);

если при некотором $j \in [1 : n]$ выполняется $\Delta_j < 0$, то перейти к шагу V.

V. Положить $j = 1$.

VI. Если $\Delta_j < 0$, то перейти к шагу VII; иначе перейти к шагу VIII.

VII. Если при всех $k \in [1 : m]$ выполняется неравенство $z_{kj} \leqslant 0$, то прекратить вычисления (в этом случае целевая функция (c, x) не ограничена (сверху) на допустимом множестве X); иначе перейти к шагу VIII.

VIII. Если $j < n$, то положить $j = j + 1$ и перейти к шагу VI; иначе перейти к шагу IX.

IX. Выбрать по заранее оговоренному правилу индекс $s \in [1 : n]$ такой, чтобы $\Delta_s < 0$. Обычно на практике за Δ_s принимают или наименьшую из отрицательных оценок Δ_j, или же отрицательную оценку Δ_j с наименьшим индексом j.

X. Вычислить отношение z_{k0}/z_{ks} для тех k, для которых $z_{ks} > 0$, и обозначить минимальное из этих отношений через θ_0.

Найти индекс $r \in [1 : m]$ такой, что $z_{rs} > 0$ и $z_{r0}/z_{rs} = \theta_0$.

XI. При каждом $j \in [1 : n]$ вычислить

$$\theta_j = z_{rj}/z_{rs}.$$

XII. Положить $z_{kj} = z_{kj}$, $k = 1, \ldots, m$; $j = 1, \ldots, n$.

XIII. Перейти к новому базису, который получается заменой вектора a^{i_r} в предыдущем базисе вектором a^s (т. е. положить $i_r = s$).

XIV. Вычислить координаты всех векторов $a^0, a^1, \ldots, a^n$ в новом базисе по основным формулам:

при $k \neq r$

$$z_{kj} = \bar{z}_{kj} - \theta_j \bar{z}_{ks}, \quad j = 0, 1, \ldots, n;$$
$$k = 1, \ldots, r-1, r+1, \ldots, m;$$

при $k = r$

$$z_{kj} = \theta_j, \quad j = 0, 1, \ldots, n.$$

XV. Перейти к новому опорному решению $x^1 = (x_1^1, \ldots, x_n^1)$:

$$x_{i_k}^1 = z_{k0}, \quad k = 1, \ldots, m,$$

остальные координаты вектора x^1 равны нулю.

XVI. Перейти к шагу III.

Теорема 1. *Если выполнены предположения 0 и задача 0 невырожденная, то через конечное число итераций процесс решения задачи 0 алгоритмом 1 закончится либо на шаге IV (в этом случае находится оптимальное решение задачи 0), либо на шаге VII (в этом случае устанавливается, что оптимального решения задачи 0 нет).*

Замечание 1. Оценки Δ_j, $j = 1, \ldots, n$, целесообразно вычислять по формуле (4.3) только на первой итерации. На последующих итерациях новую оценку Δ_j следует вычислять через старую оценку $\bar{\Delta}_j$, полученную на предыдущей итерации, по формуле

$$\Delta_j = \bar{\Delta}_j - \theta_j \bar{\Delta}_s, \quad j = 1, 2, \ldots, n.$$

2. Методы отыскания исходного базиса

А. Первый метод. Пусть исходная задача линейного программирования имеет вид:

Задача 2. Найти $\arg\max\limits_{x \in X} (c, x)$ для заданного вектора $c = (c_1, c_2, \ldots, c_n)$ и множества X, задаваемого соотношениями

$$\begin{aligned}
&\alpha_{11}x_1 + \cdots + \alpha_{1s}x_s + x_{s+1} = a_1^0;\\
&\alpha_{21}x_1 + \cdots + \alpha_{2s}x_s + x_{s+2} = a_2^0;\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&\alpha_{k1}x_1 + \cdots + \alpha_{ks}x_s + x_n = a_k^0;\\
&\alpha_{k+1,1}x_1 + \cdots + \alpha_{k+1,s}x_s = a_{k+1}^0;\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&\alpha_{m1}x_1 + \cdots + \alpha_{ms}x_s = a_m^0;\\
&x_j \geqslant 0, \quad j = 1, \ldots, n.
\end{aligned}$$

В ограничениях задачи 2 выделены переменные $x_{s+1}, \ldots, x_n$ с различными единичными векторами $a^{s+1}, \ldots, a^n$. Если таких переменных в задаче линейного программирования нет, то в задаче 2 следует положить $s = n$.

Предположение 2. Ограничения задачи 2 таковы, что $a_i^0 \geqslant 0$, $i = 1, \ldots, m$.

Для отыскания исходного базиса и опорного решения задачи 2 в ограничения этой задачи искусственно вводят $m - k$ новых переменных $y_1, y_2, \ldots, y_{m-k}$ с таким расчетом, чтобы новая система уравнений-ограничений имела полный единичный базис. Затем с помощью алгоритма 1 решают задачу линейного программирования с новыми ограничениями и со специально построенной целевой функцией. В результате решения «новой» задачи приходят либо к опорному решению исходной задачи 2, либо непосредственно к оптимальному решению задачи 2. Кроме того, если исходная задача 2 неразрешима, то это обнаруживается в результате решения «новой» задачи.

«Новая» задача является задачей линейного программирования в пространстве R^{n+m-k} (или R^{s+m}).

З а д а ч а 2′. Найти $\arg \max\limits_{(x_1,\ldots,x_n,y_1,\ldots,y_{m-k})\in Y} (-y_1-y_2-\ldots-y_{m-k})$ для множества Y, задаваемого неравенствами

$$\begin{aligned}
&\alpha_{11}x_1 + \cdots + \alpha_{1s}x_s + x_{s+1} &&= a_1^0;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\alpha_{k1}x_1 + \cdots + \alpha_{ks}x_s + x_n &&= a_k^0;\\
&\alpha_{k+1,1}x_1 + \cdots + \alpha_{k+1,s}x_s + y_1 &&= a_{k+1}^0;\\
&\alpha_{k+2,1}x_1 + \cdots + \alpha_{k+2,s}x_s + y_2 &&= a_{k+2}^0;\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\alpha_{m1}x_1 + \cdots + \alpha_{ms}x_s + \cdots + y_{m-k} &&= a_m^0;
\end{aligned}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n; \quad y_i \geqslant 0, \quad i = 1, 2, \ldots, m-k.$$

Задача 2′ имеет опорное решение $(0, \ldots, 0, a_1^0, a_2^0, \ldots, a_m^0)$ с единичным базисом $a^{s+1}, a^{s+2}, \ldots, a^n, a^{n+1}, \ldots, a^{n+m-k}$, поэтому к ней применим алгоритм 1 — симплекс-метод решения канонической задачи линейного программирования. При этом задача 2′ заведомо имеет оптимальное решение, так как ее целевая функция ограничена сверху числом ноль. Переменные $y_1, \ldots, y_{m-k}$ называют *искусственными переменными,* а векторы $a^{n+1}, \ldots, a^{n+m-k}$ — *искусственными векторами.*

Алгоритм 2

I. Используя алгоритм 1, вычислить вектор $(x_1^0, x_2^0, \ldots, x_n^0, y_1^0, \ldots$ $\ldots, y_{m-k}^0)$ — оптимальное решение задачи 2′ и оптимальный базис задачи 2′ — $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$.

II. Если хотя бы при одном $\bar{j} \in [1 : m-k]$ выполняется неравенство $y_{\bar{j}} > 0$, то прекратить вычисления (в этом случае исходная задача 2 недопустима, т. е. не имеет допустимых решений); если $y_1^0 = y_2^0 = \ldots = y_{m-k}^0 = 0$, то перейти к шагу III.

III. Если оптимальный базис задачи 2′ не содержит искусственных векторов a^j, $j = n + 1, \ldots, n + m - k$, то прекратить вычисления, так как оптимальный базис задачи 2′ является исходным базисом задачи 2, а соответствующие ему значения основных переменных $x_1^0, x_2^0, \ldots, x_n^0$ составляют исходное опорное решение задачи 2; если оптимальный базис задачи 2′ содержит хотя бы один искусственный вектор a^j, $j \in [n + 1 : n + m - k]$, то перейти к шагу IV.

IV. Вектор $(x_1^0, x_2^0, \ldots, x_n^0)$ является опорным решением задачи 2. Для отыскания исходного базиса задачи 2 перейти к шагу V.

V. Если в разложении векторов $a^1, a^2, \ldots, a^n$ по базису $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ все коэффициенты z_{kj} при искусственных векторах равны нулю, то, удалив из системы векторов $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ искусственные векторы, получим исходный базис задачи 2, который соответствует опорному решению $(x_1^0, \ldots, x_n^0)$; иначе перейти к шагу VI.

VI. Найти вектор a^s, $(s \in [1 : n])$, в разложении которого по базису $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ есть число z_{rs} при искусственном векторе a^{i_r}, отличное от нуля. Ввести в базис вектор a^s вместо искусственного вектора a^{i_r} (т. е. положить $i_r = s$).

VII. Если в новом базисе нет искусственных векторов, то прекратить вычисления (в этом случае находится опорное решение задачи 2 и его базис); иначе перейти к шагу VIII.

VIII. Разложить по новому базису векторы $a^1, a^2, \ldots, a^n$ (как на шаге XIV алгоритма 1).

IX. Если в разложении векторов $a^1, a^2, \ldots, a^n$ по базису $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ все коэффициенты z_{kj} при искусственных векторах равны нулю, то перейти к шагу V; иначе перейти к шагу VI.

Теорема 2. *При выполнении предположений 2 алгоритм 2 за конечное число итераций либо находит опорное решение задачи 2 и соответствующий ему базис, состоящий из l векторов (l — ранг матрицы A задачи 2), либо устанавливает, что задача 2 не имеет допустимых решений.*

Б. Второй метод. Ниже приводится алгоритм отыскания опорного решения задачи 0 и его базиса, который сводится к решению M-задачи — специально построенной задачи линейного программирования в пространстве R^{n+m}, содержащей параметр M (здесь M — достаточно большое число). В процессе решения M-задачи либо устанавливается неограниченность целевой функции задачи 0, либо находится оптимальное решение задачи 0, либо устанавливается, что задача 0 не имеет допустимых решений.

Предположение 2′. Ограничения задачи 0 таковы, что $a_i^0 \geqslant 0$, $i = 1, 2, \ldots, m$.

Алгоритм 2′

I. Выбрать достаточно большое число $M > 0$.

II. Используя алгоритм 1, вычислить вектор $(x_1^0, \ldots, x_n^0, y_1^0, \ldots$

$\ldots, y_m^0)$ — оптимальное решение следующей задачи линейного программирования:

М-задача. Найти

$$\arg \max_{(x_1, \ldots, x_n, y_1, \ldots, y_m)} [(c, x) - My_1 - My_2 - \cdots - My_m]$$

при ограничениях

$$\alpha_{11}x_1 + \alpha_{12}x_2 + \cdots + \alpha_{1n}x_n + y_1 = a_1^0;$$
$$\alpha_{21}x_1 + \alpha_{22}x_2 + \cdots + \alpha_{2n}x_n + y_2 = a_2^0;$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\alpha_{m1}x_1 + \alpha_{m2}x_2 + \cdots + \alpha_{mn}x_n + y_m = a_m^0;$$
$$x_j \geqslant 0, \quad j = 1, \ldots, n; \quad y_i \geqslant 0, \quad i = 1, \ldots, m.$$

M-задача имеет исходное опорное решение $(0, \ldots, 0, a_1^0, \ldots, a_m^0)$ с единичным базисом $a^{n+1}, \ldots, a^{n+m}$, поэтому к ней можно применить алгоритм 1. Если при решении M-задачи алгоритмом 1 окажется, что ее целевая функция неограничена (сверху) на допустимом множестве, то целевая функция задачи 0 также не ограничена (сверху) на допустимом множестве X.

III. Если при некотором $i_0 \in [1 : m]$ выполняется $y_{i_0}^0 > 0$, то прекратить вычисления (в этом случае задача 0 не имеет допустимых решений); если при всех $i \in [1 : m]$ выполняется $y_i^0 = 0$, то положить $x^* = (x_1^0, \ldots, x_n^0)$ и прекратить вычисления (т. е. в этом случае вектор $(x_1^0, x_2^0, \ldots, x_n^0)$ является оптимальным решением задачи 0).

Теорема 2′. *Пусть выполняется предположение 2′. Если для всех достаточно больших значений M M-задача имеет оптимальное решение, то алгоритм 2′ либо устанавливает недопустимость задачи 0, либо находит оптимальное ее решение, а если при тех же условиях M-задача неразрешима, то неразрешима и задача 0.*

3. Симплекс-метод решения вырожденной канонической задачи линейного программирования

В этом пункте приводится симплекс-метод решения вырожденной задачи 0, для которой выполняются условия (*i*), (*ii*), (*iii*) предположения 0.

Если применить алгоритм 1 для решения вырожденной задачи 0, то возможно зацикливание процесса решения задачи (т. е. возвращение к уже пройденному базису бесконечное число раз), так как на шаге X алгоритма 1 в вырожденном случае существует, вообще говоря, больше чем один номер r, для которого достигается минимальное отношение $\theta_0 = z_{r0}/z_{rs}$. Симплекс-метод для решения вырожденной задачи линейного программирования включает в себя специальное правило выбора выводимого вектора из базиса, которое гарантирует от зацикливания процесса решения задачи.

Алгоритм 3

I—IX (шаги I—IX такие, как и в алгоритме 1).

X. Вычислить отношение z_{k0}/z_{ks} для тех k, для которых $z_{ks} > 0$ и обозначить минимальное из этих отношений через θ_0.

XI. Положить $j = 0$.

XII. Положить $\theta_0^0 = \theta_0$.

XIII. Вычислить индексы $r_1, r_2, \ldots, r_{l_j}$ такие, что

$$z_{r_1 0}/z_{r_1 s} = z_{r_2 0}/z_{r_2 s} = \cdots = z_{r_{l_j} 0}/z_{r_{l_j} s} = \theta_0^0.$$

XIV. Если $l_j = 1$, то вместо вектора $a^{i_{r_1}}$ ввести в базис вектор a^s (т. е. положить $i_{r_1} = s$) и перейти к шагу XIX; если $l_j > 1$, то перейти к шагу XV.

XV. Вычислить

$$\theta_0^{j+1} = \min\{z_{r_1(j+1)}/z_{r_1 s},\ z_{r_2(j+1)}/z_{r_2 s}, \ldots, z_{r_{l_j}(j+1)}/z_{r_{l_j} s}\}.$$

XVI. Положить $\bar r_1 = r_1, \bar r_2 = r_2, \ldots, \bar r_{l_j} = r_{l_j}$.

XVII. Среди множества индексов $\{\bar r_1, \bar r_2, \ldots, \bar r_{l_j}\}$ найти индексы $r_1, r_2, \ldots, r_{l_{j+1}}$, для которых

$$z_{r_1(j+1)}/z_{r_1 s} = z_{r_2(j+1)}/z_{r_2 s} = \cdots = z_{r_{l_{j+1}}(j+1)}/z_{r_{l_{j+1}} s} = \theta_0^{j+1}.$$

XVIII. Положить $j = j + 1$ и перейти к шагу XIV.

XIX. При каждом $j \in [1 : n]$ вычислить

$$\theta_j = z_{r_1 j}/z_{r_1 s}.$$

XX. Положить $\bar z_{kj} = z_{kj}$, $k = 1, \ldots, m$; $j = 1, \ldots, n$.

XXI. Вычислить координаты векторов $a^0, a^1, \ldots, a^n$ в новом базисе:

при $k \neq r_1$

$$z_{kj} = \bar z_{kj} - \theta_j \bar z_{ks}, \quad j = 0, 1, \ldots, n;$$
$$k = 1, \ldots, r_1 - 1, r_1 + 1, \ldots, m;$$

при $k = r_1$

$$z_{kj} = \theta_j, \quad j = 0, 1, \ldots, n.$$

XXII. Перейти к новому опорному решению $x^1 = (x_1^1, \ldots, x_n^1)$:

$$x_{i_k}^1 = z_{k0}, \quad k = 1, \ldots, m,$$

остальные координаты вектора x^1 равны нулю.

XXIII. Перейти к шагу III.

Теорема 3. *Если выполнены условия (i), (ii), (iii) предположения 0, то через конечное число итераций процесс решения задачи 0 алгоритмом 3 закончится либо на шаге IV (находится оптимальное решение задачи 0, равное опорному $x^1 = (x_1^1, \ldots, x_n^1)$), либо на шаге VII (устанавливается, что оптимального решения задачи 0 нет).*

Замечание 3. На практике алгоритм 3 используется редко, поскольку он требует значительно больше времени для решения задачи линейного программирования по сравнению с алгоритмом 1, а зацикливание процесса решения вырожденной задачи алгоритмом 1 маловероятно. Если же при решении задачи 0 алгоритмом 1 произошло зацикливание на некотором опорном решении, то следует использовать алгоритм 3 для получения нового опорного решения и дальше продолжать решение задачи 0 с помощью алгоритма 1.

4. Модифицированный симплекс-метод

Во многих случаях модифицированный симплекс-метод (или метод обратной матрицы) более экономный в вычислениях, чем обычный (особенно это проявляется, если число уравнений-ограничений m существенно меньше размерности пространства n). На каждой итерации в модифицированном алгоритме вычисляется матрица, обратная к базисной, что сводится к вычислению по основным формулам m векторов размерности m (напомним, что в основном симплекс-методе на каждой итерации по основным формулам вычисляется $n - m + 1$ вектор). Модифицированный симплекс-метод применяется для решения невырожденных и вырожденных канонических задач линейного программирования.

Алгоритм 4

Начало. I. Найти исходное опорное решение $x^1 = (x_1^1, x_2^1, \ldots, x_n^1)$ задачи 0 и базис $a^{l_1}, a^{l_2}, \ldots, a^{l_m}$ этого опорного решения. В общем случае для нахождения исходного опорного решения и соответствующего ему базиса используют алгоритм 2 или алгоритм 2′.

II. Вычислить матрицу $B^{-1} = (\beta_{kj})_{k=1,\ldots,m}^{j=1,\ldots,m}$, обратную к исходной базисной матрице B.

III. Положить $z_{k0} = x_{l_k}^1$, $k = 1, 2, \ldots, m$.

Основной цикл. IV. Положить $c_{\text{баз}} = (c_{l_1}, c_{l_2}, \ldots, c_{l_m})$.

V. Вычислить вектор

$$b = c_{\text{баз}} B^{-1}.$$

VI. Для каждого $j \in [1 : n]$ вычислить оценку

$$\Delta_j = ba^j - c_j, \quad j = 1, \ldots, n.$$

VII. Если все $\Delta_j \geqslant 0$ $(j = 1, \ldots, n)$, то положить $x^* = x^1$ и прекратить вычисления (т. е. в этом случае опорное решение x^1 оптимально); если при некотором $j \in [1 : n]$ выполняется $\Delta_j < 0$, то перейти к шагу VIII.

VIII. Выбрать по заранее оговоренному правилу индекс $s \in [1 : n]$ такой, чтобы $\Delta_s < 0$.

IX. Вычислить вектор-столбец

$$(z_{1s}, z_{2s}, \ldots, z_{ms})^T = B^{-1} a^s.$$

X. Если при всех $k \in [1:m]$ выполняется $z_{ks} \leqslant 0$, то прекратить вычисления (в этом случае целевая функция (c, x) не ограничена (сверху) на допустимом множестве X); иначе перейти к шагу XI.

XI. Вычислить отношение z_{k0}/z_{ks} для всех $k \in [1:m]$, для которых $z_{ks} > 0$ и обозначить минимальное из этих отношений через θ_0.

XII. Найти индекс $r \in [1:m]$ такой, что $r_{rs} > 0$ и $z_{r0}/z_{rs} = \theta_0$.

XIII. Положить $\bar{\beta}_{kj} = \beta_{kj}$, $k = 1, ..., m$; $j = 1, ..., m$, и $\bar{z}_{k0} = z_{k0}$, $k = 1, ..., m$.

XIV. Перейти к новому базису $a^{i_1}, ..., a^{i_m}$, который получается заменой вектора a^{i_r} в предыдущем базисе вектором a^s (т. е. положить $i_r = s$).

XV. Вычислить по основным формулам матрицу $B^{-1} = (\beta_{kj})_{k=1,...,m}^{j=1,...,m}$, обратную к новой базисной матрице

$$\beta_{rj} = \bar{\beta}_{rj} / z_{rs}, \quad j = 1, \ldots, m;$$

$$\beta_{kj} = \bar{\beta}_{kj} - (\bar{\beta}_{rj}/z_{rs}) z_{ks}, \quad k = 1, \ldots, m;$$

$$j = 1, \ldots, m; \quad k \neq r.$$

XVI. Вычислить по основным формулам новые значения базисных координат опорного решения

$$z_{r0} = z_{r0}/z_{rs};$$

$$z_{k0} = \bar{z}_{k0} - (\bar{z}_{r0}/z_{rs}) z_{ks}, \; k = 1, 2, \ldots, r-1, r+1, \ldots, m.$$

XVII. Перейти к новому опорному решению $x^1 = (x_1^1, ..., x_n^1)$:

$$x_{i_k}^1 = z_{k0}, \quad k = 1, \ldots, m,$$

остальные координаты вектора x^1 равны нулю.

XVIII. Перейти к шагу IV.

Теорема 4. *Если выполнены предположения 0 и задача 0 невырожденная, то процесс решения задачи 0 алгоритмом 4 через конечное число итераций закончится либо на шаге VII (находится оптимальное решение задачи 0), либо на шаге X (устанавливается, что оптимального решения задачи 0 нет).*

Замечание 4. На практике алгоритм 4 применяется также для решения вырожденной задачи линейного программирования, для которой выполняются условия (*i*), (*ii*), (*iii*) предположения 0. В этом случае теоретически возможно зацикливание процесса решения задачи 0 алгоритмом 4. Если при решении задачи 0 алгоритмом 4 на некотором опорном решении все-таки произойдет зацикливание, то (аналогично пункту 3) необходимо включить в алгоритм 4 специальное правило выбора выводимого вектора из базиса, гарантирующее от зацикливания.

Замечание 4'. Если на шаге XV алгоритма 4 матрицу B^{-1} вычислить по формуле $B^{-1} = D_r\bar{B}^{-1}$, где $\bar{B}^{-1}$ — матрица, обратная к

базисной на предыдущей итерации, а D_r имеет вид

$$D_r = \begin{Vmatrix} 1 & \ldots & 0 & -z_{1s}/z_{rs} & 0 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \ldots & 1 & -z_{r-1,s}/z_{rs} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1/z_{rs} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & -z_{r+1,s}/z_{rs} & 1 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \ldots & 0 & -z_{ms}/z_{rs} & 0 & \ldots & 1 \end{Vmatrix},$$

то такой модифицированный симплекс-метод называется *мультипликативным*. Для записи матрицы D_r достаточно знать $m+1$ чисел: число r и числа $-z_{1s}/z_{rs}$, ..., $1/z_{rs}$, ..., $-z_{ms}/z_{rs}$. Если обозначить через D_{r_k} матрицу D_r в k-й итерации мультипликативного алгоритма и за исходную базисную выбрать единичную матрицу I, то обратная к базисной в k-й итерации будет вычисляться по формуле

$$B^{-1} = D_{r_k} D_{r_{k-1}} \ldots D_{r_1} I.$$

Из этой формулы видно, что для вычисления матрицы B^{-1} в k-й итерации мультипликативного алгоритма требуется знать $(m+1)k$ чисел, что на начальной стадии вычислений приводит к экономии оперативной памяти ЭВМ.

Библиографические указания. При написании параграфа использованы работы [414, 415, 416, 226, 114]. Дополнительные сведения о симплекс-методе и его вариантах, о построении исходного опорного решения и его базиса, о решении вырожденных задач можно найти в работах [414, 415, 416, 411, 78, 65, 11, 187, 21, 45, 199, 348, 64].

4.2. Двойственный симплекс-метод

Двойственной (сопряженной) к задаче 1.0 (см. § 4.1, задача 0) называется задача 0.

Задача 0. Найти $\arg\min\limits_{y \in Y} (a^0, y)$ для заданного вектора a^0 и множества Y, заданного соотношениями

$$Y = \{y \mid A^T y \geqslant c^T, \quad y \in R^m\}.$$

Предположения 0. (*i*) — ранг матрицы A равен m; (*ii*) — $n > m$; (*iii*) — задача 1.0 разрешима.

1. Основной алгоритм

Определение 1. Почти допустимым опорным решением задачи 1.0 называется вектор $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$, удовлетворяющий уравнению $Ax = a^0$, причем векторы a^j, соответствующие ненулевым координатам вектора $\tilde{x}$, линейно-независимы.

Определение 2. Базисом почти допустимого опорного решения $\tilde{x}$ задачи 1.0 называется любой упорядоченный набор из m линейно-независимых векторов a^j, содержащий все векторы a^j, соответствующие ненулевым координатам вектора $\tilde{x}$.

Практический интерес представляет почти допустимое опорное решение и его базис, для которых справедливо $\Delta_j \geqslant 0$, $j = 1, \ldots, n$. В [415] такое решение называется псевдопланом задачи 1.0, а базис — псевдобазисом задачи 1.0. В алгоритме 1 на каждой итерации осуществляется переход от одного базиса задачи 0 к другому (или, что то же самое, от одного псевдобазиса задачи 1.0 к другому).

Определение 3. Допустимое решение y двойственной задачи 0 называется опорным, если среди условий $A^T y \geqslant c^T$, которые оно обращает в равенства, имеется m линейно-независимых условий.

Определение 4. Базисом опорного решения двойственной задачи 0 называется произвольная система из m линейно-независимых векторов $a^{i_1}, \ldots, a^{i_m}$ прямой задачи, для которых

$$(y,\ a^{i_s}) = c_{i_s}, \quad s = 1, \ldots, m.$$

Определение 5. Опорное решение y двойственной задачи 0 называется невырожденным, если для любого вектора a^j, не входящего в его базис, $(y,\ a^j) > c_j$.

Определение 6. Двойственная задача 0, все опорные решения которой являются невырожденными, называется невырожденной.

Двойственный симплекс-метод (или метод последовательного уточнения оценок) решения задачи 1.0 по своей сути — это применение обычного симплекс-метода к двойственной задаче 0, дополненное построением на каждой итерации n-мерного вектора $\tilde{x}$, являющегося почти допустимым опорным решением задачи 1.0 (отсюда происходит название метода). Двойственный симплекс-метод заключается в таком последовательном переходе от одного почти допустимого решения $\tilde{x}$ к другому, что через конечное число переходов либо получим оптимальное решение задачи 1.0, либо установим, что задача 1.0 не имеет допустимых решений. Оптимальному решению x^* задачи 1.0 соответствует оптимальное решение y^* задачи 0, причем оптимальные значения прямой и двойственной задач совпадают.

Оптимальное решение двойственной задачи определяется по формуле $y^* = (c_{i_1}, c_{i_2}, \ldots, c_{i_m})B^{-1}$, где B — базисная матрица оптимального опорного решения задачи 1.0; $i_1, \ldots, i_m$ — индексы базисных векторов.

Для начала работы алгоритма необходимо иметь опорное решение $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m)$ сопряженной (двойственной) задачи и его базис $a^{i_1}, \ldots, a^{i_m}$, на основании которых вычисляется исходное почти

допустимое опорное решение с неотрицательными оценками по правилу:

в начале находят коэффициенты разложения z_{l0}, $l = 1, \ldots, m$, вектора a^0 по базису $a^{i_1}, \ldots, a^{i_m}$; далее составляют вектор $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$, l-я компонента $\tilde{x}_{i_l}$ ($l = 1, \ldots, m$) которого равна z_{l0}, остальные компоненты вектора $\tilde{x}$ равны нулю. Алгоритмы отыскания опорного решения сопряженной задачи и его базиса приведены в пункте 2.

Алгоритм 1 применяют для вырожденной и невырожденной задачи 1.0. Если при использовании алгоритма к решению задачи 1.0 в вырожденном случае возникает зацикливание, то в алгоритме 1 необходимо применять процедуру, гарантирующую от зацикливания.

Алгоритм 1

Начало. I. Найти почти допустимое опорное решение $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n)$ задачи 1.0 (с неотрицательными оценками Δ_j, $j = 1, 2, \ldots, n$) и базис $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ этого почти допустимого опорного решения.

II. Вычислить матрицу B^{-1}, обратную к исходной базисной матрице B, состоящей из столбцов $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$.

III. Разложить по исходному базису $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ векторы a^0, $a^1, a^2, \ldots, a^n$ (т. е. найти числа z_{kj}, $k = 1, 2, \ldots, m$; $j = 0, 1, \ldots, n$, такие, что $a^j = \sum_{k=1}^{m} z_{kj} a^{i_k}$, $j = 0, 1, \ldots, n$), по формулам

$$z_{10} = \tilde{x}_{i_1}, \quad z_{20} = \tilde{x}_{i_2}, \ \ldots, \ z_{m0} = \tilde{x}_{i_m};$$

$$(z_{1j}, z_{2j}, \ldots, z_{mj})^T = B^{-1}(\alpha_{1j}, \alpha_{2j}, \ldots, \alpha_{mj})^T, \ j = 1, \ldots, n.$$

Отметим, что заранее известно разложение по исходному базису базисных векторов a^{i_k}, $k = 1, \ldots, m$:

$$z_{li_k} = 0 \text{ при } k \neq l, \ z_{li_k} = 1 \text{ при } k = l, \ k = 1, \ldots, m;$$

$$l = 1, \ldots, m.$$

IV. Вычислить для каждого $j \in [1 : n]$ оценку

$$\Delta_j = \sum_{k=1}^{m} c_{i_k} z_{kj} - c_j.$$

Основной цикл. V. Если для всех $k \in [1 : m]$ выполняется неравенство $z_{k0} \geqslant 0$, то положить $x^* = \tilde{x}$ и прекратить вычисления (т. е. в этом случае почти допустимое опорное решение $\tilde{x}$ оптимально); иначе перейти к шагу VI.

VI. Если существует такое $k \in [1 : m]$, что $z_{k0} < 0$ и при всех $j \in [1 : n]$ выполняется $z_{kj} \geqslant 0$, то прекратить вычисления (в этом

случае задача 1.0 не имеет допустимых решений); иначе перейти к шагу VII.

VII. Выбрать (по заранее оговоренному правилу) индекс $r \in [1 : m]$ такой, чтобы $z_{r0} < 0$.

VIII. Вычислить отношения Δ_j/z_{rj} для всех тех $j \in [1 : n]$, для которых $z_{rj} < 0$ и обозначить минимальное по абсолютной величине из этих отношений через θ_0 (напомним, что $\Delta_j \geqslant 0$, $j \in [1 : n]$).

IX. Найти индекс $s \in [1 : n]$ такой, что $z_{rs} < 0$ и $-\Delta_s/z_{rs} = \theta_0$. Для невырожденной задачи 1.0 индекс s определяется однозначно. В вырожденном случае минимальное отношение θ_0 может достигаться для нескольких индексов $j \in [1 : n]$. Поэтому на практике в качестве s выбирается наименьший индекс j, для которого достигается минимальное отношение. Теоретически в вырожденном случае необходимо применять процедуру выбора s, гарантирующую от зацикливания (см. пункт 3).

X. Положить $\bar{z}_{kj} = z_{kj}$, $k = 1, \ldots, m$; $j = 1, \ldots, n$ и $\bar{\Delta}_j = \Delta_j$, $j = 1, \ldots, n$.

XI. Перейти к новому базису $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$, который получается заменой вектора a^{i_r} в предыдущем базисе вектором a^s (т. е. положить $i_r = s$).

XII. Вычислить координаты векторов $a^0, a^1, \ldots, a^n$ в новом базисе:

$$z_{kj} = \bar{z}_{kj} - (\bar{z}_{rj}/\bar{z}_{rs})\,\bar{z}_{ks}, \quad j = 0, 1, \ldots, n;$$
$$k = 1, \ldots, r-1, r+1, \ldots, m.$$
$$z_{rj} = \bar{z}_{rj}/\bar{z}_{rs}, \quad j = 0, 1, \ldots, n.$$

XIII. Вычислить по основным формулам оценки Δ_j, $j \in [1 : n]$, отвечающие новому базису

$$\Delta_j = \bar{\Delta}_j - (\bar{z}_{rj}/\bar{z}_{rs})\,\bar{\Delta}_s, \quad j = 1, \ldots, n.$$

XIV. Перейти к новому почти допустимому опорному решению

$$\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n);$$
$$\tilde{x}_{i_k} = z_{k0}, \quad k = 1, 2, \ldots, m,$$

остальные координаты вектора $\tilde{x}$ равны нулю.

XV. Перейти к шагу V.

Теорема 1. *Если выполнены предположения 0 и задача 0 невырожденная, то процесс решения задачи 1.0 с помощью алгоритма 1 закончится через конечное число итераций, либо на шаге V (в этом случае находится оптимальное решение задачи 1.0), либо на шаге VI (в этом случае устанавливается, что задача 1.0 не имеет допустимых решений).*

Замечание 1. Так как число итераций, необходимых для решения задачи линейного программирования, определяется в основном

количеством непрямых ограничений, то двойственный симплекс-метод целесообразно применять, когда число непрямых ограничений существенно превосходит число переменных.

2. Методы отыскания исходного опорного решения сопряженной задачи

В подпункте 2′ дается метод отыскания опорного решения сопряженной задачи 0 при условии, что известно произвольное допустимое решение задачи 0. В 2″ на основании метода, приведенного в 2′, рассматривается алгоритм, позволяющий найти оптимальное решение сопряженной задачи (а значит, и прямой) без предварительного вычисления допустимого решения сопряженной задачи.

2′. Метод отыскания опорного решения сопряженной задачи по известному допустимому решению. Предварительно укажем класс задач, в которых определение допустимого решения не представляет труда. Пусть в задаче 1.0 одна и та же компонента α_{tj}, $j = 1, \ldots, n$, всех векторов a^j, $j = 1, \ldots, n$, положительна. Построить вектор

$$y' = \underbrace{(\overbrace{0, \ldots, 0, y_t}^{t}, 0, \ldots, 0)}_{m},$$

где $y_t \geqslant \max\limits_{1 \leqslant j \leqslant n} (c_j/\alpha_{tj})$.

Вектор y' удовлетворяет ограничениям задачи 0 и поэтому является допустимым решением задачи 0.

Предположение 2′. (*i*) — ранг матрицы A равен m; (*ii*) — $n > m$; (*iii*) — допустимое множество сопряженной задачи 0 непусто.

Алгоритм 2

I. Найти вектор $y' = (y_1', y_2', \ldots, y_m')$ — допустимое решение сопряженной задачи 0 такое, что

$$\sum_{i=1}^{m} \alpha_{ij} y_i' > c_j, \quad j = 1, \ldots, n_1; \tag{4.4}$$

$$\sum_{i=1}^{m} \alpha_{ij} y_i' = c_j, \quad j = n_1 + 1, \ldots, n. \tag{4.5}$$

II. Найти число r — количество линейно-независимых условий в (4.5). Если $r = m$, то прекратить вычисления (т. е. y' — опорное решение задачи 0, а векторы $a^{i_1}, \ldots, a^{i_m}$, определяемые линейно-независимыми условиями, составляют базис этого опорного решения); иначе перейти к шагу III.

III. Выразить из системы равенств

$$\sum_{i=1}^{m} \alpha_{ij} y_i = c_j, \quad j = n_1 + 1, \ldots, n, \tag{4.6}$$

r переменных $y_1, \ldots, y_r$ через остальные переменные:

$$y_j = \sum_{i=r+1}^{m} d_{ij} y_i + d_j, \quad j = 1, \ldots, r. \qquad (4.7)$$

IV. Заменить переменные y_i, $i = 1, \ldots, r$, в неравенствах

$$\sum_{i=1}^{m} \alpha_{ij} y_i > c_j, \quad j = 1, \ldots, n_1,$$

их выражениями из (4.7) и получить систему неравенств относительно $y_{r+1}, \ldots, y_m$

$$\sum_{i=r+1}^{m} \alpha'_{ij} y_i > c'_j, \quad j = 1, \ldots, n_1. \qquad (4.8)$$

V. Подставить в линейную форму (a^0, y) задачи 0 значения переменных $y_1, \ldots, y_r$ из (4.7) и получить линейную функцию g_1 переменных $y_{r+1}, \ldots, y_m$

$$g_1(y_{r+1}, \ldots, y_m) = \sum_{i=r+1}^{m} b_i y_i + b_0.$$

VI. Если $b_i = 0$, $i = r + 1, \ldots, m$, то перейти к шагу XIII; иначе перейти к шагу VII.

VII. Вычислить

$$\mu_j = \sum_{i=r+1}^{m} \alpha'_{ij} b_i, \quad j = 1, \ldots, n_1.$$

VIII. Если $\mu_j \leqslant 0$, $j = 1, \ldots, n_1$, то прекратить вычисления (в этом случае линейная форма задачи 0 не ограничена снизу на допустимом множестве); иначе перейти к шагу IX.

IX. Вычислить

$$\Delta_j = \sum_{i=r+1}^{m} \alpha'_{ij} y'_i - c'_j, \quad j = 1, \ldots, n_1. \qquad (4.9)$$

X. Вычислить

$$\theta_0 = \min_{\mu_j > 0} (\Delta_j / \mu_j).$$

XI. Перейти к новому допустимому решению $y'' = (y''_1, \ldots, y''_m)$ по правилам

$$y''_i = y'_i - \theta_0 b_i, \quad i = r + 1, \ldots, m;$$

$y''_1, \ldots, y''_r$ находят через $y''_{r+1}, \ldots, y''_m$ по (4.7).

XII. Вычислить индекс $k \in [1 : n_1]$ такой, что

$$\theta_0 = (\Delta_k / \mu_k), \quad \mu_k > 0,$$

и прекратить вычисления.

В этом случае построено новое допустимое решение y'', которое обращает в равенства $r + 1$ линейно-независимых условий из

системы ограничений задачи 0, т. е. r линейно-независимых условий из системы (4.5)

$$\sum_{i=1}^{m} \alpha_{ij} y''_i = c_j, \quad j = n_1 + 1, \ \ldots, \ n,$$

и линейно-независимое от системы (4.5) условие из (4.4)

$$\sum_{i=1}^{m} \alpha_{ik} y''_i = c_k, \quad k \in [1 : n_1].$$

XIII. Вычислить величины Δ_j, $j = 1, \ldots, n_1$ по (4.9).

XIV. Вычислить величину θ_0 по правилу

$$\theta_0 = \begin{cases} \min\limits_{\alpha'_{r+1,j}>0} (\Delta_j/\alpha'_{r+1,j}), & \text{если среди } \alpha'_{r+1,j},\ j = 1, \ldots, n_1, \\ & \text{имеются положительные величины;} \\ -\min\limits_{\alpha'_{r+1,j}<0} (-\Delta_j/\alpha'_{r+1,j}), & \text{если } \alpha'_{r+1,j} \leqslant 0 \text{ для } j = 1, \ldots, n_1. \end{cases}$$

XV. Перейти к новому допустимому решению $y'' = (y''_1, \ldots, y''_m)$ по правилам:

$$y''_i = y'_i, \quad i = r + 2, \ \ldots, \ m;$$

$$y''_{r+1} = y'_{r+1} - \theta_0,$$

$y''_1, \ldots, y''_r$ находят через $y''_{r+1}, \ldots, y''_m$ по формуле (4.7).

XVI. Вычислить индекс $k \in [1 : n_1]$ такой, что

$$\theta_0 = \Delta_k/\alpha'_{r+1,k}$$

и прекратить вычисления (см. шаг XII).

Теперь можно применить алгоритм 2 к допустимому решению y'' (т. е. сделать еще одну итерацию) и получить допустимое решение y''', которое превращает в равенства $r + 2$ линейно-независимых ограничений задачи 0 и т. д. Не более чем за $m - r'$ итераций будет получено опорное решение сопряженной задачи 0 (здесь r' — число линейно-независимых условий-равенств для y').

2″. Метод отыскания оптимального решения сопряженной задачи без предварительного вычисления допустимого решения.

Алгоритм 2′

I. Выбрать достаточно большое число $\beta > 0$ (β больше любого числа, с которым его приходится по ходу решения сравнивать).

II. Составить расширенную сопряженную задачу:

$$\text{найти } \arg\min \left(\beta y_0 + \sum_{i=1}^{m} a_i^0 y_i\right) \text{ при ограничениях}$$

$$y_0 + \sum_{i=1}^{m} \alpha_{ij} y_i \geqslant c_j, \quad j = 1, \ \ldots, \ n;$$

$$y_0 \geqslant 0.$$

(Прямой расширенной задачей является задача 1.0 (с переменными $x_0, x_1, \ldots, x_n$), в которой к ограничениям $Ax = a^0$, $x \geqslant 0$ добавлены ограничения $x_0 + x_1 + \ldots + x_n = \beta$; $x_0 \geqslant 0$).

III. Вычислить

$$y'_0 = \max\{c_1, \ldots, c_n, 0\}.$$

IV. Составить вектор

$$y' = \underbrace{(y'_0, 0, \ldots, 0)}_{m+1},$$

являющийся допустимым решением расширенной сопряженной задачи.

V. Отправляясь от допустимого решения y' с помощью метода, приведенного в подпункте 2′, вычислить опорное решение расширенной сопряженной задачи, его базис и почти допустимое опорное решение расширенной прямой задачи.

VI. Решить прямую расширенную задачу двойственным симплекс-методом и получить векторы: $\bar{x}^* = (x_0^*, x_1^*, \ldots, x_n^*)$ — оптимальное решение прямой расширенной задачи; $\bar{y}^* = (y_0^*, y_1^*, \ldots, y_m^*)$ — оптимальное решение расширенной сопряженной задачи.

VII. Если $y_0^* = 0$, то $y^* = (y_1^*, \ldots, y_m^*)$ — оптимальное решение сопряженной задачи 0, а $x^* = (x_1^*, \ldots, x_n^*)$ — оптимальное решение задачи 1.0.

Если $y_0^* > 0$, то сопряженная задача 0 и прямая задача 1.0 неразрешимы.

Алгоритм 2′ за конечное число итераций находит оптимальное решение прямой задачи 1.0 и двойственной задачи 0 без предварительного вычисления допустимого решения сопряженной задачи.

Недостатком алгоритма 2′ является проблема выбора подходящего числа $\beta > 0$.

3. Правило выбора вектора a^s, вводимого в базис, гарантирующее от зацикливания в вырожденном случае

Если при использовании алгоритма 1 для решения задачи линейного программирования в вырожденном случае возникло зацикливание (т. е. возврат к уже пройденному базису), то на шаге IX алгоритма 1 следует применить следующую процедуру выбора индекса $s \in [1 : n]$, гарантирующую от зацикливания.

Алгоритм 3

Начало. I. Обозначить через $\mathcal{J}_1$ совокупность индексов j, на которых достигается

$$\theta_0 = \min_{z_{rj} < 0} (-\Delta_j / z_{rj}).$$

II. Положить $\nu = 1$.

Основной цикл. III. Если $\mathcal{J}_\nu$ содержит один элемент, то этот элемент принять в качестве индекса s и прекратить вычисления; иначе перейти к шагу IV.

IV. Если все индексы $j \in \mathcal{J}_\nu$ превосходят $i_{m-\nu+1}$, то положить $s = \min\limits_{j \in \mathcal{J}_\nu} j$; иначе перейти к шагу V.

V. Построить множество $\mathcal{J}_{\nu+1}$, объединяя в него те индексы j, на которых достигается

$$\min_{\substack{j \in \mathcal{J}_\nu \\ j < i_{m-\nu+1}}} (z_{m-\nu+1,j}/z_{rj}).$$

VI. Если $\nu < m$, то положить $\nu = \nu + 1$ и перейти к шагу III; иначе перейти к шагу VII.

VII. Положить $s = \min\limits_{j \in \mathcal{J}_{m+1}} j$.

Библиографические указания. Пункт 1 написан на основании работы [226], пункты 2 и 3 — на основании работ [414, 415, 416].

Дополнительные сведения о двойственном симплекс-методе, способах построения исходного опорного решения и вырожденности в двойственном симплекс-методе можно найти в работах [81, 429, 226, 114].

4.3. Методы последовательного сокращения невязок

Процесс вычисления оптимального решения методом сокращения невязок не требует предварительного определения опорного решения прямой или двойственной задачи. Для начала процесса достаточно знать лишь произвольное допустимое решение сопряженной задачи. При наличии опорного решения исходной задачи 1.0 и произвольного допустимого решения сопряженной задачи 2.0 модификация метода сокращения невязок — метод двухсторонних оценок позволяет получать приближенное решение с наперед заданной ошибкой при существенно меньшем числе итераций, чем это требуется для получения оптимального решения.

1. Метод последовательного сокращения невязок

Предположения 1. (i) — $a_i^0 \geqslant 0$, $i = 1, 2, \ldots, m$; (ii) — ранг матрицы A равен m; (iii) — $n > m$; (iv) — двойственная задача 2.0 имеет допустимые решения.

Метод последовательного сокращения невязок использует прямую и двойственную задачи линейного программирования. Правила перехода от одной итерации к другой обеспечивают сокращение невязок (разностей между правыми и левыми частями условий-равенств задачи 1.0). Через конечное число итераций либо невязки сводятся к нулю (т. е. находится оптимальное решение задачи 1.0), либо устанавливается неразрешимость задачи 1.0. На каждой итерации алгоритма требуется вычислять оптимальные решения некоторой вспомогательной задачи и двойственной к ней. В методе

требуется предварительное вычисление допустимого решения двойственной задачи 2.0.

Алгоритм 1

Начало. I. Найти произвольное допустимое решение $y = (y_1, y_2, \ldots, y_m)$ двойственной задачи 2.0.

Основной цикл. II. Найти все индексы $j \in [1:n]$, для которых

$$\sum_{i=1}^{m} \alpha_{ij} y_i = c_j,$$

и обозначить множество таких индексов через

$$\mathcal{J} \triangleq \{j_1, j_2, \ldots, j_l\}.$$

III. Используя модифицированный симплекс-метод, найти оптимальное решение $(\bar{x}^*_{j_1}, \bar{x}^*_{j_2}, \ldots, \bar{x}^*_{j_l}, \varepsilon^*_1, \varepsilon^*_2, \ldots, \varepsilon^*_m)$ вспомогательной задачи 1 линейного программирования в пространстве R^{l+m}:

Задача 1. Найти $\arg \min\limits_{(\bar{x}_{j_1}, \ldots, \bar{x}_{j_l}, \varepsilon_1, \ldots, \varepsilon_m)} \sum\limits_{i=1}^{m} \varepsilon_i$ при ограничениях

$$\sum_{j \in \mathcal{J}} \alpha_{ij} \bar{x}_j + \varepsilon_i = a^0_i, \quad i = 1, \ldots, m;$$

$$\bar{x}_j \geqslant 0 \text{ при } j \in \mathcal{J};$$

$$\varepsilon_i \geqslant 0, \quad i = 1, 2, \ldots, m.$$

IV. Найти оптимальное решение $\mu^* = (\mu^*_1, \mu^*_2, \ldots, \mu^*_m)$ задачи, двойственной к приведенной выше вспомогательной задачи 1, которое можно вычислить по формуле

$$\mu^* = (\bar{c}_{i_1}, \bar{c}_{i_2}, \ldots, \bar{c}_{i_m}) B^{-1}_{\mathcal{J}},$$

где $B^{-1}_{\mathcal{J}}$ — матрица, обратная к оптимальной базисной матрице задачи 1; $(\bar{c}_{i_1}, \ldots, \bar{c}_{i_m})$ — координаты вектора $(\underbrace{0, \ldots, 0}_{l}, \underbrace{1, 1, \ldots, 1}_{m})$, отвечающие оптимальному базису задачи 1. При решении задачи 1 модифицированным симплекс-методом матрица $B^{-1}_{\mathcal{J}}$ и вектор $(\bar{c}_{i_1}, \bar{c}_{i_2}, \ldots, \bar{c}_{i_m})$ известны.

Двойственная к задаче 1 имеет вид:

найти $\arg\max\limits_{(\mu_1, \ldots, \mu_m)} \sum\limits_{i=1}^{m} a^0_i \mu_i$ при ограничениях

$$\sum_{i=1}^{m} \alpha_{ij} \mu_i \leqslant 0 \text{ при } j \in \mathcal{J};$$

$$\mu_i \leqslant 1, \quad i = 1, 2, \ldots, m.$$

V. Если невязки ε_i^*, $i = 1, 2, \ldots, m$, удовлетворяют равенствам

$$\varepsilon_i^* = 0, \quad i = 1, 2, \ldots, m,$$

то вычислить вектор $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ по правилу

$$\left.\begin{aligned} x_j^* &= \bar{x}_j^* \ \text{при } j \in \mathcal{J} \\ x_j^* &= 0 \ \text{при } j \notin \mathcal{J} \end{aligned}\right\}, \quad j = 1, \ldots, n,$$

и прекратить вычисления (в этом случае находится оптимальное решение x^* задачи 1.0);

если при некотором $i \in [1:m]$ $\varepsilon_i^* > 0$, то перейти к шагу VI.

VI. Вычислить величины

$$\delta_j^* = \sum_{i=1}^{m} \alpha_{ij}\mu_i^*, \quad j = 1, \ldots, n.$$

VII. Если при всех $j \in [1:n]$ выполняется неравенство $\delta_j^* \leqslant 0$, то прекратить вычисления (в этом случае задача 1.0 не имеет допустимых решений); иначе перейти к шагу VIII.

VIII. Вычислить величины Δ_j для тех j, для которых $\delta_j^* > 0$:

$$\Delta_j = \sum_{i=1}^{m} \alpha_{ij}y_i - c_j.$$

IX. Вычислить отношение Δ_j/δ_j^* для тех j, для которых $\delta_j^* > 0$, и минимальное из этих отношений обозначить через θ_0.

X. Положить $\bar{y} = y$.

XI. Перейти к новому допустимому решению $y = (y_1, \ldots, y_m)$ двойственной задачи 2.0 по правилу

$$y_i = \bar{y}_i - \theta_0\mu_i^*, \quad i = 1, \ldots, m.$$

XII. Перейти к шагу II.

Теорема 1. *Если выполнены предположения 1 и расширенная задача 1 невырожденная, то процесс решения задачи 1.0 алгоритмом 1 через конечное число итераций закончится либо на шаге V (находится оптимальное решение задачи 1.0), либо на шаге VII (устанавливается, что задача 1.0 не имеет допустимых решений).*

Замечание 1. Алгоритм 1 можно применять для решения задачи 1.0 и в том случае, когда расширенная задача 1 вырожденная. Для этого необходимо при решении вспомогательных задач модифицированным симплекс-методом пользоваться правилом, гарантирующим от зацикливания.

Замечание 1'. Исходное опорное решение каждой последующей вспомогательной задачи 1 следует составлять из соответствующих положительных компонент оптимального решения предыдущей вспомогательной задачи 1, что существенно сократит число итераций, необходимых для решения очередной вспомогательной задачи 1.

2. Метод двусторонних оценок

Для решения задач линейного программирования методом двусторонних оценок предварительно необходимо находить (кроме исходного допустимого решения сопряженной задачи 2.0) исходное опорное решение исходной задачи 1.0. На каждой итерации алгоритма вычисляется некоторая оценка (невязка ε) уклонения полученного опорного решения от оптимального. В методе двусторонних оценок требуется на каждой итерации решать вспомогательную задачу линейного программирования с известным исходным опорным решением и его базисом.

Алгоритм 2

Начало. I. Найти опорное решение $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ задачи 1.0 и допустимое решение $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m)$ сопряженной задачи 2.0.

II. Вычислить величины

$$\Delta_j = \sum_{i=1}^{m} a_{ij}\bar{y}_i - c_j, \quad j \in [1:n].$$

Основной цикл. III. Найти все индексы $j \in [1:n]$, для которых выполняется $\Delta_j = 0$, и обозначить множество таких индексов через $\mathcal{J}_Y$.

IV. Обозначить через $\mathcal{J}_X$ множество индексов векторов базиса опорного решения $\bar{x}$.

V. Найти множество

$$\mathcal{J} = \mathcal{J}_Y \cup \mathcal{J}_X.$$

VI. Используя модифицированный симплекс-метод, найти оптимальное решение $\bar{x}^*$ следующей вспомогательной задачи линейного программирования.

Задача 2. Найти $\arg\min\limits_{x_j} \sum\limits_{j \in \mathcal{J}} \Delta_j x_j$ при ограничениях

$$\sum_{j \in \mathcal{J}} a^j x_j = a^0;$$

$$x_j \geqslant 0, \quad j \in \mathcal{J}.$$

VII. Найти оптимальное решение $\mu^* = (\mu_1^*, \ldots, \mu_m^*)$ задачи, двойственной к задаче 2

$$\mu^* = (\Delta_{i_1}, \ldots, \Delta_{i_m}) B_{\mathcal{J}}^{-1},$$

где $B_{\mathcal{J}}^{-1}$ — матрица, обратная к оптимальной базисной матрице задачи 2; $i_1, \ldots, i_m$ — номера базисных векторов оптимального решения задачи 2.

VIII. Вычислить невязку

$$\varepsilon = \sum_{j \in \mathcal{J}} \Delta_j \bar{x}^*.$$

Невязка ε является разностью между значениями линейных форм двойственной и прямой задач при данных $\bar{x}$ и $\bar{y}$. На практике вычисления можно прекращать, когда ε не превышает наперед заданной точности вычисления минимума целевой функции.

IX. Если $\varepsilon = 0$, то вычислить оптимальное решение $x^* = (x_1^*, \ldots$ $\ldots, x_n^*)$ задачи 1.0

$$\left.\begin{aligned} x_j^* &= \bar{x}_j^* \text{ при } j \in \mathcal{J} \\ x_j^* &= 0 \text{ при } j \notin \mathcal{J} \end{aligned}\right\}, \quad j = 1, \ldots, n,$$

и прекратить вычисления; иначе перейти к шагу X.

X. Вычислить параметры

$$\delta_j^* = \sum_{i=1}^{m} \alpha_{ij}\mu_i^*, \quad j = 1, 2, \ldots, n.$$

Если при всех $j \in [1:n]$ выполняется $\delta_j^* \leqslant 0$, то прекратить вычисления (в этом случае задача 1.0 не имеет допустимых решений); иначе перейти к шагу XI.

XI. Вычислить отношение Δ_j/δ_j^* для тех j, для которых $\delta_j^* >$ > 0, и минимальное из этих отношений обозначить через θ_0.

XII. Перейти к новому опорному решению $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ задачи 1.0

$$\left.\begin{aligned} \bar{x}_j &= \bar{x}_j^* \text{ при } j \in \mathcal{J} \\ \bar{x}_j &= 0 \text{ при } j \notin \mathcal{J} \end{aligned}\right\}, \quad j = 1, \ldots, n.$$

XIII. Положить $y = \bar{y}$; $\bar{\Delta}_j = \Delta_j$, $j = 1, \ldots, n$.

XIV. Перейти к новому допустимому решению $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_m)$ сопряженной задачи 2.0

$$\bar{y} = y - \theta_0\mu^*.$$

XV. Вычислить величины Δ_j, отвечающие новому допустимому решению $\bar{y}$,

$$\Delta_j = \bar{\Delta}_j - \theta_0\delta_j^*, \quad j \in [1:n].$$

XVI. Перейти к шагу III.

Теорема 2. *Если выполнены предположения 1, то в невырожденном случае процесс решения задачи 1.0 алгоритмом 2 через конечное число итераций закончится либо на шаге IX (находится оптимальное решение задачи 1.0), либо на шаге X (устанавливается, что задача 1.0 не имеет допустимых решений).*

Алгоритм 2 может применяться и для вырожденного случая. Тогда при решении вспомогательной задачи 2 теоретически может возникнуть зацикливание. При зацикливании следует применять специальное правило, гарантирующее от зацикливания.

Библиографические указания. Параграф написан на основании работ [415, 416, 116]. Метод последовательного сокращения невязок и его варианты изучались также в работах [414, 207, 241, 374].

4.4. Обобщенный симплекс-метод

Задача 1. Найти $\arg\max_{x \in X} \sum_{j=1}^{n} c_j x_j$ для заданного вектора $c = (c_1, \ldots, c_n)$ и заданного множества

$$X \triangleq \{x \mid Ax = a^0, \quad x \geqslant 0, \quad x \in R^n\}.$$

Предположения 1. (*i*) — ранг матрицы A равен m; (*ii*) — $m < n$; (*iii*) — множество X непусто.

В обобщенном симплекс-методе на каждой итерации алгоритма в базисе меняются не один, а несколько векторов. Для определения векторов, вводимых в базис и выводимых из него, требуется на каждой итерации решать вспомогательную задачу линейного программирования с ограничениями в виде неравенств. В начальной стадии алгоритма требуется иметь опорное решение, его базис и матрицу, обратную к базисной. На последующих итерациях опорное решение, его базис и матрица, обратная к базисной, вычисляются по рекуррентным формулам. На начальных итерациях алгоритма число вводимых в базис векторов надо выбирать по возможности большим, а в конце решения задачи это число следует уменьшать.

Обобщенный симплекс-метод требует меньшего, по сравнению с другими методами, числа итераций для получения оптимального решения, и важно, что этот метод менее чувствителен к ошибкам вычислений.

Алгоритм 1

Начало. I. Найти исходное опорное решение $x^1 = (x_1^1, \ldots, x_n^1)$ и его базис $a^{i_1}, \ldots, a^{i_m}$.

Способы вычисления исходного опорного решения и его базиса приведены в § 4.1.

II. Вычислить матрицу $B_{x^1}^{-1} = (\beta_{lj})_{l=1,\ldots,m}^{j=1,\ldots,m}$, обратную к исходной базисной матрице $B_{x^1} = (a^{i_1}, \ldots, a^{i_m})$.

Основной цикл. III. Положить $z_{k0} = x_{i_k}^1$, $k = 1, \ldots, m$; $\mathcal{J}_{x^1} = \{i_1, \ldots, i_m\}$.

IV. Вычислить величины

$$\lambda_l = \sum_{k=1}^{m} c_{i_k} \beta_{kl}, \quad l = 1, \ldots, m.$$

V. Вычислить оценки Δ_j, $j = 1, \ldots, n$,

$$\Delta_j = \sum_{l=1}^{m} \alpha_{lj} \lambda_l - c_j \text{ для } j \notin \mathcal{J}_{x^1};$$

$$\Delta_j = 0 \text{ для } j \in \mathcal{J}_{x^1}.$$

VI. Если все $\Delta_j \geqslant 0$ ($j = 1, \ldots, n$), то положить $x^* = x^1$ и прекратить вычисления (т. е. в этом случае опорное решение x^1 оптимально); иначе перейти к шагу VII.

VII. Задать множество L, состоящее из произвольных l индексов $j \notin \mathcal{I}_x$, таких, что $\Delta_j < 0$.

VIII. Вычислить коэффициенты z_{ij}, $i = 1, \ldots, m$; $j \in L$, разложения векторов a^j, $j \in L$ по базисным векторам

$$z_{ij} = \sum_{k=1}^{m} \alpha_{kj}\beta_{ik}, \quad i = 1, \ldots, m; \quad j \in L.$$

IX. Применить алгоритм 7.1 (см. § 4.7, алгоритм 1) к решению следующей вспомогательной задачи линейного программирования: найти $\arg\max_{\theta_l} \left(-\sum_{l \in L} \Delta_l \theta_l\right)$ для заданных чисел Δ_l, $l \in L$ при ограничениях

$$z_{i0} - \sum_{l \in L} z_{il}\theta_l \geqslant 0, \quad i = 1, \ldots, m;$$

$$\theta_l \geqslant 0, \quad l \in L.$$

За исходное опорное решение вспомогательной задачи можно принимать $\theta_l = 0$, $l \in L$ (базис этого опорного решения не содержит ни одного элемента). Решение вспомогательной задачи может завершиться либо признаком неразрешимости этой задачи, либо получением оптимального решения, базиса этого решения и матрицы, обратной к оптимальной базисной (см. § 4.7).

Если в процессе решения вспомогательной задачи обнаружится ее неразрешимость, то прекратить вычисления (в этом случае задача 1 также неразрешима, т. е. ее целевая функция неограничена на допустимом множестве X); иначе получить оптимальное решение θ_l^*, $l \in L$, вспомогательной задачи и его базис $\bar{B}_\tau$, $(1 \leqslant \tau \leqslant t)$, состоящий из элементов матрицы $(z_{il})_{i=1,\ldots,m}^{l \in L}$, которые расположены на пересечении строк и столбцов с номерами $r_1, \ldots, r_\tau$ и $k_1, \ldots, k_\tau$, соответственно, а также получить матрицу $\bar{B}_\tau^{-1}$, обратную к матрице $\bar{B}_\tau$.

X. Перейти к новому опорному решению $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ по правилам

$$\bar{x}_j = \begin{cases} z_{k0} - \sum_{\mu \in L} z_{k\mu}\theta_\mu^*, & \text{если } j = i_k, \ k = 1, \ldots, m; \\ \theta_j^*, & j \in L, \\ 0, & \text{в остальных случаях,} \end{cases}$$

где $j = 1, \ldots, n$.

XI. Перейти к новому базису $B_{\bar{x}}$ путем замены в старом базисе векторов $a^{i_{r_\lambda}}$, $\lambda = 1, \ldots, \tau$, векторами $a^{k_1}, \ldots, a^{k_\tau}$, т. е. положить $i_{r_\lambda} = k_\lambda$, $\lambda = 1, \ldots, \tau$.

XII. Получить из матрицы $B_{\bar{x}}$ путем перестановок ее столбцов матрицу $A_{\bar{x}}$ по следующему правилу:

первые τ столбцов матрицы $A_{\bar{x}}$ образуются матрицей $C_1 = (a^{k_1}, a^{k_2}, \ldots, a^{k_\tau})$; остальные $m - \tau$ столбцов матрицы образуются матрицей C_2, которая получается из матрицы $B_{\bar{x}}$ путем вычеркивания столбцов с индексами $k_1, \ldots, k_\tau$. Указанный способ получения матрицы $A_{\bar{x}}$ из $B_{\bar{x}}$ порождает набор чисел s_i, $i = 1, \ldots, m$, ($s_i \in [1 : m]$) такой, что i-й (в порядке следования) столбец матрицы $B_{\bar{x}}$ ставится на место s_i-го столбца матрицы $A_{\bar{x}}$. Обозначить через Q_1 ($Q_1(C) = \bar{C}$) оператор, который i-й ($i = 1, \ldots, m$) столбец квадратной матрицы C ставит на место s_i-го ($i = 1, \ldots, m$) столбца в матрице $\bar{C}$, и обозначить через Q_2 ($Q_2(D) = \bar{D}$) оператор, который i-ю ($i = 1, \ldots, m$) строку квадратной матрицы D ставит на место s_i-й строки в матрице $\bar{D}$.

XIII. Составить матрицу Y размера $m \times \tau$, столбцами которой являются векторы

$$z^{k_\lambda} = (z_{1k_\lambda}, z_{2k_\lambda}, \ldots, z_{mk_\lambda})^T, \quad \lambda = 1, \ldots, \tau.$$

XIV. Составить матрицу $\bar{\bar{B}}_{m-\tau,\tau}$, которая получается из матрицы Y путем вычеркивания строк с индексами $r_1, r_2, \ldots, r_\tau$.

XV. Разбить матрицу B_x^{-1} на две подматрицы

$$(\beta_{ij})_{i=1,\ldots,\tau}^{j=1,\ldots,m} \text{ и } (\beta_{i+\tau,j})_{i=1,\ldots,m-\tau}^{j=1,\ldots,m}$$

размера $\tau \times m$ и $(m - \tau) \times m$, соответственно.

XVI. Вычислить матрицы

$$(\bar{\beta}_{i+\tau,j})_{i=1,\ldots,m-\tau}^{j=1,\ldots,m} \text{ и } (\bar{\beta}_{ij})_{i=1,\ldots,\tau}^{j=1,\ldots,m},$$

соответственно, по формулам

$$(\bar{\beta}_{i+\tau,j})_{i=1,\ldots,m-\tau}^{j=1,\ldots,m} = (\beta_{i+\tau,j})_{j=1,\ldots,m-\tau}^{i=1,\ldots,m} - \bar{\bar{B}}_{m-\tau,\tau}\bar{B}_\tau^{-1}(\beta_{ij})_{i=1,\ldots,\tau}^{j=1,\ldots,m};$$

$$(\bar{\beta}_{ij})_{i=1,\ldots,\tau}^{j=1,\ldots,m} = \bar{B}_\tau^{-1}(\beta_{ij})_{i=1,\ldots,\tau}^{j=1,\ldots,m}.$$

XVII. Составить матрицу $A_{\bar{x}}^{-1}$ размера $m \times m$, первые τ строк которой образованы матрицей $(\bar{\beta}_{ij})_{i=1,\ldots,\tau}^{j=1,\ldots,m}$, а остальные $m - \tau$ — матрицей $(\bar{\beta}'_{i+\tau,j})_{i=1,\ldots,m-\tau}^{j=1,\ldots,m}$.

XVIII. Найти матрицу

$$B_{\bar{x}}^{-1} = Q_2^{-1}(A_{\bar{x}}^{-1}).$$

XIX. Положить $x = \bar{x}$, $B_x^{-1} = B_{\bar{x}}^{-1}$, $B_x = B_{\bar{x}}$ и перейти к шагу III.

Теорема 1. *Если выполнены предположения 1, то в невырожденном случае процесс решения задачи 1 алгоритмом 1 через конечное число итераций закончится либо на шаге VI (находится оптимальное решение*

x^* задачи 1), либо на шаге IX (устанавливается, что целевая функция задачи 1 неограничена на допустимом множестве X).

Библиографические указания. При написании параграфа использованы работы [414, 415, 416, 114].

4.5. Методы блочного программирования

Методы блочного программирования позволяют получить решение экстремальной задачи с помощью решения ряда вспомогательных задач, каждая из которых в определенном смысле является более простой, чем исходная. В линейном программировании методы блочного программирования применяются для решения задач большого размера и задач с блочной структурой матрицы ограничений.

1. Метод разложения Данцига—Вульфа

Задача 1. Найти $\arg\max_{x \in X} (c, x)$ для заданного вектора $c = (c_1, c_2, \ldots, c_n)$ и множества X, задаваемого соотношениями

$$A^0 x = b^0; \qquad (4.10)$$

$$A^1 x = b^1; \qquad (4.11)$$

$$x \geqslant 0, \qquad (4.12)$$

где A^0 — матрица размера $m \times n$; A^1 — матрица размера $m_1 \times n$; $b^0 = (b_1, \ldots, b_m)$; $b^1 = (b_{m+1}, \ldots, b_{m+m_1})$.

1′. Случай ограниченности множества X^1. *Предположения 1.* (*i*) — множество X^1, определяемое условиями (4.11) — (4.12) — ограничено; (*ii*) — $\operatorname{rang}\begin{pmatrix} A^0 \\ A^1 \end{pmatrix} = m + m_1$; (*iii*) — $m + m_1 < n$.

В методах разложения исходная задача 1 размера $(m + m_1) \times n$ сводится к решению следующей задачи линейного программирования размера $(m + 1) \times l$.

z-задача. Найти $\arg\max_z \sum_{i=1}^{l} \sigma_i z_i$ при ограничениях

$$\sum_{i=1}^{l} p^i z_i = b^0; \qquad (4.13)$$

$$\sum_{i=1}^{l} z_i = 1; \qquad (4.14)$$

$$z_i \geqslant 0, \quad i = 1, 2, \ldots, l, \qquad (4.15)$$

где $x^1, x^2, \ldots, x^l$ — вершины многогранника X^1;

$$\sigma_i = (c, x^i), \quad i = 1, 2, \ldots, l; \quad p^i = A^0 x^i, \quad i = 1, 2, \ldots, l. \qquad (4.16)$$

Оптимальное решение $(z_1^*, z_2^*, \ldots, z_l^*)$ *z*-задачи однозначно определяет оптимальное решение x^* задачи 1

$$x^* = \sum_{i=1}^{l} z_i^* x^i.$$

Для решения z-задачи не обязательно знать все вершины множества X^1 (т. е. вектора $x^1, x^2, \ldots, x^l$). Необходимо лишь знать вершины, входящие в исходный опорный базис z-задачи. В процессе решения z-задачи модифицированным симплекс-методом проверка на оптимальность опорного решения z-задачи производится с помощью решения некоторой вспомогательной задачи линейного программирования с ограничениями (4.11) — (4.12).

Алгоритм 1

Начало. I. Найти исходное опорное решение $z^1 = (z_1^1, \ldots \ldots, z_l^1)$ z-задачи, базис $\begin{pmatrix} p^{i_1} \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} p^{i_{m+1}} \\ 1 \end{pmatrix}$ этого опорного решения и коэффициенты σ_{i_k}, $k = 1, \ldots, m+1$, линейной формы z-задачи, отвечающие исходному опорному решению.

II. Вычислить матрицу B^{-1}, обратную матрице, составленной из базисных векторов $\begin{pmatrix} p^{i} \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} p^{i_{m+1}} \\ 1 \end{pmatrix}$

$$B^{-1} \triangleq (\beta_{ij})_{i=1,\ldots,m+1}^{j=1,\ldots,m+1}.$$

Основной цикл. III. Положить $\sigma_{баз} = (\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_{m+1}})$.

IV. Вычислить вектор

$$\bar{\Lambda} \triangleq (\Lambda, \lambda) = (\lambda_1, \lambda_2, \ldots, \lambda_m, \lambda)$$

по формуле

$$\bar{\Lambda} = \sigma_{баз} B^{-1}.$$

V. Положить $y_{k0} = z_{i_k}^1$, $k = 1, 2, \ldots, m+1$.

VI. Вычислить n-мерный вектор $c^\Lambda = (c_1^\Lambda, c_2^\Lambda, \ldots, c_n^\Lambda)$ по формуле

$$c^\Lambda = c - \Lambda A^0$$

(в обычной форме записи координаты вектора c^Λ вычисляются по формуле

$$c_j^\Lambda = c_j - \sum_{i=1}^{m} \lambda_i \alpha_{ij}^0, \quad j = 1, 2, \ldots, n).$$

VII. Вычислить вектор x^ν — оптимальное решение следующей задачи линейного программирования в пространстве R^n. Подзадача X_Λ: найти $\arg\max_x (c^\Lambda, x)$ при ограничениях

$$A^1 x = b^1;$$

$$x \geqslant 0.$$

Для решения подзадачи X_Λ удобно применять симплекс-метод или модифицированный симплекс-метод. Причем на каждой последующей итерации за исходное опорное решение и его базис можно брать оптимальное решение и его базис, полученные на предыдущей итерации.

VIII. Если выполняется равенство $(c^\Lambda, x^\nu) = \lambda$, то положить $z^* = z^1$ и перейти к шагу IX (т. е. в этом случае опорное решение $z^1 = (z_1^1, \ldots, z_l^1)$ является и оптимальным решением z-задачи); иначе перейти к шагу X.

IX. Вычислить вектор x^* — оптимальное решение задачи 1

$$x^* = \sum_{k=1}^{m+1} z_{i_k}^* x^{i_k}$$

и прекратить вычисления.

X. Вычислить вектор $\bar{p}^\nu = \begin{pmatrix} p^\nu \\ 1 \end{pmatrix}$, где

$$p^\nu = A^0 x^\nu.$$

XI. Вычислить вектор (т. е. разложить вектор $\bar{p}^\nu$ по базисным векторам)

$$(y_{1\nu}, y_{2\nu}, \ldots, y_{m+1,\nu})^T = B^{-1}\bar{p}^\nu.$$

XII. Вычислить коэффициент линейной формы z-задачи

$$\sigma_\nu = (c, x^\nu).$$

XIII. Вычислить отношения $y_{k0}/y_{k\nu}$ для тех $k \in [1 : (m+1)]$, для которых $y_{k\nu} > 0$ и обозначить минимальное из этих отношений через θ_0.

XIV. Найти индекс $r \in [1 : (m+1)]$ такой, что $y_{r\nu} > 0$ и $y_{r0}/y_{r\nu} = \theta_0$.

XV. Положить $\bar{\beta}_{ij} = \beta_{ij}$, $i = 1, \ldots, m+1$; $j = 1, \ldots, m+1$ и $\bar{y}_{k0} = y_{k0}$, $k = 1, \ldots, m+1$.

Перейти к новому базису $\begin{pmatrix} p^{i_1} \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} p^{i_r} \\ 1 \end{pmatrix}, \ldots, \begin{pmatrix} p^{i_{m+1}} \\ 1 \end{pmatrix}$, который получается заменой вектора $\bar{p}^{i_r} = \begin{pmatrix} p^{i_r} \\ 1 \end{pmatrix}$ в предыдущем базисе вектором $\bar{p}^\nu = \begin{pmatrix} p^\nu \\ 1 \end{pmatrix}$ (т. е. положить $i_r = \nu$).

XVI. Вычислить по основным формулам матрицу $B^{-1} \triangleq (\beta_{ij})_{i=1,\ldots,m+1}^{j=1,\ldots,m+1}$, обратную к новой базисной матрице

$$\beta_{rj} = \bar{\beta}_{rj}/y_{r\nu}, \quad j = 1, \ldots, m+1;$$

$$\beta_{ij} = \bar{\beta}_{ij} - (\bar{\beta}_{rj}/\bar{\beta}_{r\nu})\, y_{i\nu},$$

$$i = 1, \ldots, m+1; \quad j = 1, \ldots, m+1; \quad i \neq r.$$

XVII. Вычислить по основным формулам новые значения базисных координат опорного решения

$$y_{r0} = \bar{y}_{r0}/y_{r\nu};$$

$$y_{k0} = \bar{y}_{k0} - (\bar{y}_{r0}/y_{r\nu})\, y_{k\nu}, \quad k = 1, \ldots, r-1, r+1, \ldots, m+1.$$

XVIII. Перейти к новому опорному решению z-задачи

$$z^1 = (z_1^1, \ldots, z_l^1):$$

$$z_{i_k}^1 = y_{k0}, \quad k = 1, \ldots, m+1,$$

остальные координаты вектора z^1 равны нулю.

XIX. Перейти к шагу III.

Теорема 1. *Если допустимое множество решений X задачи 1 непусто, то при выполнении предположений 1 алгоритм 1 за конечное число итераций приводит к оптимальному решению задачи 1 (предполагается, что если в вырожденном случае возникает зацикливание, то по аналогии с пунктом 3 § 4.1 применяется специальное правило выхода из цикла).*

1″. Случай неограниченности множества X^1. *Предположения 1′* (*i*).— Множество X^1, определяемое условиями (4.11) — (4.12) — неограничено; (*ii*) — rang $\begin{pmatrix} A^0 \\ A^1 \end{pmatrix} = m + m_1$; (*iii*) — $m + m_1 < n$.

Обозначим через x^i, $i = 1, \ldots, l_1$, вершины множества X^1, а направляющие векторы неограниченных ребер X^1 через x^{l_1+1}, x^{l_1+2}, ..., x^l.

Если выполняются предположения 1′, то исходная задача 1 сводится к решению z-задачи, в которой условие (4.14) следует заменить на

$$\sum_{i=1}^{l} \varepsilon_i z_i = 1,$$

где $\varepsilon_i = 1$ при $i = 1, 2, \ldots, l_1$; $\varepsilon_i = 0$ при $i = l_1 + 1, \ldots, l$. К измененной z-задаче можно применить алгоритм 1 со следующими изменениями: поскольку на шаге VII алгоритма 1 подзадача X_Λ не всегда разрешима (целевая функция может быть неограниченной на множестве X^1), то возможны два случая:

1°) оптимальное решение x^ν подзадачи X_Λ существует;

2°) при решении подзадачи X_Λ есть оценка $\Delta_j < 0$, для которой все $u_{kj} \leqslant 0$, $k = 1, \ldots, m_1$ (т. е. целевая функция подзадачи X_Λ не ограничена сверху).

В случае 1° шаги алгоритма 1 останутся без изменений. В случае 2° шаги I—VI останутся без изменений, а на шаге VII алгоритма 1 в качестве вектора x^ν выбирается n-мерный вектор, у которого координаты с номерами $j_1, j_2, \ldots, j_{m_1}$ равны, соответственно, $-u_{1j}$, $-u_{2j}$, ..., $-u_{m_1j}$, j-я координата равна единице, а остальные координаты нулевые (здесь $j_1, j_2, \ldots, j_{m_1}$ — номера базисных векторов подзадачи X_Λ) и осуществляется переход к шагу X, на котором вычисляется вектор $\bar{p}^\nu$ по измененной формуле

$$\bar{p}^\nu = \begin{pmatrix} p^\nu \\ 0 \end{pmatrix},$$

где

$$p^\nu = A^0 x^\nu.$$

В случае 2° вектор x^v является направляющим вектором неограниченного ребра многогранника X^1 (здесь u_{kj}, $k = 1, \ldots, m_1$ — коэффициенты разложения j-го вектора матрицы ограничений в подзадаче X_λ).

Шаг XI алгоритма 1 следует дополнить проверкой на неограниченность целевой функции z-задачи.

XI. Вычислить вектор

$$(y_{1v}, \ldots, y_{m+1,v})^T = B^{-1}\bar{p}^v.$$

Если при всех $k \in [1 : (m+1)]$ выполняется $y_{kv} \leqslant 0$, то прекратить вычисления (в этом случае целевая функция z-задачи (а значит, и задачи 1) неограничена на допустимом множестве); иначе перейти к шагу XII.

Остальные шаги алгоритма 1 остаются без изменений.

Замечание 1. Принцип разложения, в свою очередь, можно применить и для решения вспомогательной задачи на шаге VII алгоритма 1 и т. д. Поэтому решение задач линейного программирования с любым числом ограничений в принципе можно свести к решению ряда задач линейного программирования, количество ограничений в каждой из которых не превышает заданного числа m.

Теорема 1'. *Если допустимое множество решений X задачи 1 непусто, то при выполнении предположения 1' процесс решения задачи 1 измененным алгоритмом 1 через конечное число итераций либо закончится на шаге IX (находится оптимальное решение задачи 1), либо на шаге XI (устанавливается, что целевая функция задачи 1 неограничена на допустимом множестве).*

2. Метод разложения решения задач линейного программирования с блочно-диагональной матрицей

Метод разложения, описанный в предыдущем пункте, особенно эффективен при специальной блочно-диагональной структуре матрицы A^1 в задаче 1:

$$A^1 = \begin{pmatrix} A_1 & 0 & \ldots & 0 \\ 0 & A_2 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & A_s \end{pmatrix},$$

где $A_t = (a_{ij}^t)$, $t = 1, \ldots, s$ — матрица размера $m_t \times n_t$. Исходная задача линейного программирования записывается в виде.

Задача 2. Найти $\arg\max\limits_{x} \sum\limits_{t=1}^{s} (c^t, x^t)$ при ограничениях

$$\sum_{t=1}^{s} A_t^0 x^t = b^0, \tag{4.17}$$

$$A_t x^t = b^t, \quad t = 1, 2, \ldots, s, \tag{4.18}$$

$$x^t \geqslant 0, \quad t = 1, 2, \ldots, s, \tag{4.19}$$

где x^t — n_t-мерный вектор-столбец; $A_t^0 = (\alpha_{ij}^{t0})$ — матрица размера $m \times n_t$; $A_t = (\alpha_{ij}^t)$ — матрица размера $m_t \times n_t$; b^0 — m-мерный вектор-столбец; b^t — m_t-мерный вектор-столбец; c^t — n_t-мерный вектор-строка; $x = (x^1, x^2, \ldots, x^s)$.

Решение задачи 2 сводится к решению следующей задачи линейного программирования размера $(m+s) \times \hat{l}\left(\hat{l} = \sum_{t=1}^{s} l_t\right)$:

z_1-задача. Найти $\arg\max\limits_{z^t} \sum_{t=1}^{s} \sum_{i=1}^{l_t} \sigma_i^t z_i^t$ при ограничениях

$$\sum_{t=1}^{s} \sum_{i=1}^{l_t} p_{(i)}^t z_i^t = b^0; \tag{4.20}$$

$$\sum_{i=1}^{l_t} \varepsilon_i^t z_i^t = 1, \quad t = 1, \ldots, s; \tag{4.21}$$

$$z_i^t \geqslant 0, \quad i = 1, 2, \ldots, l_t; \quad t = 1, 2, \ldots, s, \tag{4.22}$$

где $\sigma_i^t = (c^t, x_{(i)}^t)$; $p_{(i)}^t = A_t^0 x_{(i)}^t$; $x_{(i)}^t$, $i = 1, \ldots, l_t$ при каждом $t \in [1 : s]$ — соответствует вершинам и направляющим векторам неограниченных ребер многогранного множества G_t, $G_t \triangleq \{x^t \mid A_t x^t = b^t,\ x^t \geqslant 0\}$, (предполагается, что при $i \in [1 : l_t^1]$, $x_{(i)}^t$ соответствует вершине, а при $i \in [(l_t^1 + 1) : l_t]$ — направляющим векторам неограниченных ребер G_t);

$$\varepsilon_i^t = \begin{cases} 1, & \text{если } x_{(i)}^t \text{ соответствует вершине множества } G_t, \\ 0, & \text{если } x_{(i)}^t \text{ соответствует бесконечному ребру множества } G_t. \end{cases}$$

Для удобства записи ограничения (4.20) и (4.21) перепишем в виде

$$\sum_{t=1}^{s} \sum_{i=1}^{l_t} \bar{p}_{(i)}^t z_i^t = \bar{b}^0,$$

где

$$\bar{p}_{(i)}^t = \begin{pmatrix} p_{(i)}^t \\ \varepsilon_i^t e^t \end{pmatrix}, \quad \bar{b}^0 = \begin{pmatrix} b_0 \\ I_s \end{pmatrix},$$

здесь e^t — s-мерный вектор с единицей на t-м месте; I_s — s-мерный вектор с единичными компонентами.

Оптимальное решение z_1-задачи однозначно определяет оптимальное решение задачи 2, т. е. если числа $\overset{*}{z}{}_i^t$ определяют решение z_1-задачи, то векторы

$$\overset{*}{x}{}^t = \sum_{i=1}^{l_t} \overset{*}{z}{}_i^t x_{(i)}^t, \quad t = 1, \ldots, s, \tag{4.23}$$

составляют оптимальное решение

$$x^* = (\overset{*}{x}{}^1, \overset{*}{x}{}^2, \ldots, \overset{*}{x}{}^s) \tag{4.24}$$

задачи 2.

Для решения z_1-задачи нет необходимости знать заранее все вершины многогранников G_t, $t = 1, \ldots, s$ (необходима лишь информация о вершинах, входящих в исходный базис). z_1-задача распадается на s (по числу блоков) подзадач размера $m_t \times n_t$, $t = 1, \ldots, s$, каждая из которых решается модифицированным симплекс-методом. Оптимальные значения линейных форм этих подзадач используются для проверки опорного решения z_1-задачи на оптимальность и для обновления базиса z_1-задачи, если исследуемое опорное решение не оптимально.

Алгоритм 2

Начало. I. Найти исходный базис z_1-задачи

$$\bar{p}^{t_1}_{(i_1)},\ \bar{p}^{t_2}_{(i_2)},\ \ldots,\ \bar{p}^{t_{m+s}}_{(i_{m+s})},$$

базисные переменные опорного решения z_1-задачи, отвечающие этому базису,

$$z^{t_1}_{i_1},\ z^{t_2}_{i_2},\ \ldots,\ z^{t_{m+s}}_{i_{m+s}}$$

и коэффициенты линейной формы z_1-задачи, отвечающие исходному базису,

$$\sigma^{t_1}_{i_1},\ \sigma^{t_2}_{i_2},\ \ldots,\ \sigma^{t_{m+s}}_{i_{m+s}}.$$

II. Вычислить матрицу B^{-1}, обратную к матрице, составленной из базисных векторов $\bar{p}^{t_k}_{(i_k)}$, $k = 1, \ldots, m + s$:

$$B^{-1} = (\beta_{ij})^{j=1,\ldots,s+m}_{i=1,\ldots,m+s}.$$

Основной цикл. III. Положить $\sigma_{\text{баз}} = (\sigma^{t_1}_{i_1}, \ldots, \sigma^{t_{m+s}}_{i_{m+s}})$.

IV. Вычислить вектор $\bar{\Lambda} \triangleq (\Lambda^1, \Lambda^2) = (\lambda^1_1, \ldots, \lambda^1_m, \lambda^2_1, \ldots, \lambda^2_s)$ по формуле

$$\bar{\Lambda} = \sigma_{\text{баз}} B^{-1}.$$

V. Положить

$$y_{k0} = z^{t_k}_{i_k}, \quad k = 1, 2, \ldots, m + s.$$

VI. Положить $t = 1$.

VII. Решить следующую задачу линейного программирования размера $m_t \times n_t$ ($X^{(t)}_\Lambda$-задачу):

найти $\arg\max\limits_{x^t} (c^t - \Lambda^1 A^0_t, x^t)$ при ограничениях

$$A_t x^t = b^t; \quad x^t \geqslant 0$$

(при этом считаются известными вершины $x^{t_1}_{i_1}, x^{t_2}_{i_2}, \ldots, x^{t_{m+s}}_{i_{m+s}}$.

Процесс решения $X_\Lambda^{(t)}$-задачи может закончиться одним из двух исходов: исход 1°— $X_\Lambda^{(t)}$-задача разрешима; исход 2°— линейная форма $X_\Lambda^{(t)}$-задачи неограничена на множестве G_t). Если $X_\Lambda^{(t)}$-задача разрешима, то обозначить ее решение через $\overset{t}{x}_{(\nu_t)}$ и перейти к шагу VIII; если $X_\Lambda^{(t)}$-задача неразрешима (т. е. линейная форма $X_\Lambda^{(t)}$-задачи неограничена на множестве G_t), то обозначить через $\overset{t}{x}_{(\nu_t)}$ направляющий вектор неограниченного ребра множества G_t, порождающий вектор $\overline{p}^t_{(\nu_t)}$ с отрицательной оценкой $\Delta^t_{\nu_t}$, и перейти к шагу IX.

(Правило выбора такого $\overset{t}{x}_{(\nu_t)}$ описано в пункте 1).

VIII. Вычислить оценку

$$\Delta^t_{\nu_t} = -(c^t - \Lambda^1 A^0_t,\ \overset{t}{x}_{(\nu_t)}) + \lambda_t^2$$

и перейти к шагу X.

IX. Вычислить оценку $\Delta^t_{\nu_t}$ по формуле

$$\Delta^t_{\nu_t} = -(c^t - \Lambda^1 A^0_t,\ \overset{t}{x}_{(\nu_t)})$$

и перейти к шагу X.

X. Если $t < s$, то положить $t = t + 1$ и перейти к шагу VII; иначе перейти к шагу XI.

XI. Если выполняется условие

$$\min_{t \in [1:s]} \Delta^t_{\nu_t} = 0, \tag{4.25}$$

то исследуемое опорное решение и базис z_1-задачи оптимальны; в этом случае найти оптимальное решение исходной задачи 2 по (4.23) — (4.24), где $\overset{*t}{z_i}$ равны соответствующим базисным переменным z_1-задачи при $t = t_k$, $i = i_k$, $k = 1, \ldots, m + s$, и $\overset{*t}{z_i} = 0$, в противном случае (векторы $\overset{t}{x_i}$, соответствующие номерам базисных переменных, тоже известны), и прекратить вычисления. Если условие (4.25) не выполняется, то перейти к шагу XII.

На последующих шагах алгоритма осуществляется переход к новому базису z_1-задачи.

XII. Вычислить индекс $\mu \in [1 : s]$, для которого выполняется

$$\Delta^\mu_{\nu_\mu} = \min_{t \in [1:s]} \Delta^t_{\nu_t}.$$

XIII. Вычислить $(m + s)$-мерный вектор $\overline{p}^\mu_{(\nu_\mu)}$ и соответствующий коэффициент линейной формы z_1-задачи

$$\overline{p}^\mu_{(\nu_\mu)} = \begin{pmatrix} A^0_\mu \cdot x^\mu_{(\nu_\mu)} \\ \varepsilon^\mu_{\nu_\mu} \cdot e^\mu \end{pmatrix}; \quad \sigma^\mu_{\nu_\mu} = (c^\mu,\ x^\mu_{(\nu_\mu)}).$$

XIV. Вычислить вектор (т. е. вычислить коэффициенты разло-

жения вектора $\bar{p}^{\mu}_{(\nu_\mu)}$ по базисным векторам)

$$(y_{1\mu}, \ldots, y_{m+s,\mu})^T = B^{-1}\bar{p}^{\mu}_{(\nu_\mu)}.$$

XV. Вычислить индекс $r \in [1 : (m+s)]$, удовлетворяющий условию

$$y_{r0}/y_{r\mu} = \min_{y_{k\mu}>0} (y_{k0}/y_{k\mu}), \quad y_{r\mu} > 0.$$

XVI. Положить $\bar{\beta}_{ij} = \beta_{ij}$, $i = 1, \ldots, m+s$; $j = 1, \ldots, m+s$, и $\bar{y}_{k0} = y_{k0}$, $k = 1, \ldots, m+s$.

XVII. Перейти к новому базису $\bar{p}^{t_1}_{(i_1)}, \ldots, \bar{p}^{t_{m+s}}_{(i_{m+s})}$, который получается заменой вектора $\bar{p}^{t_r}_{(i_r)}$ в предыдущем базисе вектором $\bar{p}^{\mu}_{(\nu_\mu)}$ (т. е. положить $t_r = \mu$, $i_r = \nu_\mu$).

Отметим, что при переходе к новому базису меняется также коэффициент линейной формы $\sigma^{t_r}_{i_r}$ и точка $x^{t_r}_{i_r}$.

XVIII. Вычислить по основным формулам матрицу B^{-1}, обратную к новой базисной матрице:

$$\beta_{rj} = \bar{\beta}_{rj}/y_{r\mu}, \quad j = 1, \ldots, m+s;$$

$$\beta_{ij} = \bar{\beta}_{ij} - (\bar{\beta}_{rj}/\bar{\beta}_{r\mu})\, y_{i\mu},$$

$$i = 1, \ldots, m+s; \quad j = 1, \ldots, m+s; \quad i \neq r.$$

XIX. Вычислить по основным формулам новые значения базисных координат опорного решения z_1-задачи:

$$y_{r0} = \bar{y}_{r0}/y_{r\mu};$$

$$y_{k0} = \bar{y}_{k0} - (\bar{y}_{r0}/y_{r\mu})\, y_{k\mu}, \quad k = 1, \ldots, r-1, r+1, \ldots, m+s.$$

XX. Положить $z^{t_k}_{i_k} = y_{k0}$, $k = 1, \ldots, m+s$, и перейти к шагу III.

Теорема 2. *Если допустимое множество решений задачи 2 непусто, уравнения-ограничения задачи 2 линейно-независимы и целевая функция задачи 2 ограничена на допустимом множестве решений, то за конечное число итераций алгоритм 2 приводит к оптимальному решению задачи 2 (при этом предполагается, что если в вырожденном случае возникает зацикливание, то, по аналогии с пунктом 3 (§ 4.1), применяется специальное правило выхода из цикла).*

3. Метод, использующий обобщенный градиентный спуск

Задача 3. Найти $\arg\max_x (c, x)$ для заданного вектора $c \in R^n$ при ограничениях

$$\sum_{j=1}^{n} \alpha^1_{ij} x_j \leq b^1_i, \quad i = 1, \ldots, m_1; \qquad (4.26)$$

$$\sum_{j=1}^{n} \alpha_{kj}^2 x_j \leqslant b_k^2, \quad k=1, \ldots, m_2; \tag{4.27}$$

$$x_j \geqslant 0, \quad j=1, \ldots, n. \tag{4.28}$$

Задача, двойственная к задаче 3, имеет вид.

Задача 3′. Найти $\arg\min\limits_{(y,z)} \left(\sum\limits_{i=1}^{m_1} b_i^1 y_i + \sum\limits_{k=1}^{m_2} b_k^2 z_k\right)$ при ограничениях

$$\sum_{i=1}^{m_1} \alpha_{ij}^1 y_i + \sum_{k=1}^{m_2} \alpha_{kj}^2 z_k \geqslant c_j, \quad j=1, \ldots, n;$$

$$y_i \geqslant 0, \quad i=1, \ldots, m_1; \quad z_k \geqslant 0, \quad k=1, \ldots, m_2,$$

здесь $y = (y_1, \ldots, y_{m_1})$, $z = (z_1, \ldots, z_{m_2})$.

Предположения 3. (*i*) — задача 3 имеет оптимальное решение; (*ii*) — система ограничений (4.27) и (4.28) вырезает в пространстве ограниченный многогранник X_1.

В приводимом ниже методе решение задачи 3′ сводится к минимизации (методом обобщенного градиентного спуска) выпуклой кусочно-линейной функции

$$\varphi^*(y) \triangleq \max_{x \in X_1} \left(\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m_1} y_i \left(b_i^1 - \sum_{j=1}^{n} \alpha_{ij}^1 x_j\right)\right)$$

при простейших ограничениях $y_i \geqslant 0$, $i = 1, \ldots, m_1$. На каждой итерации алгоритма требуется решать задачу линейного программирования с ограничениями (4.27) и (4.28).

Метод обобщенного градиентного спуска удобно применять к таким задачам линейного программирования, двойственные к которым имеют блочную структуру.

Алгоритм 3

Начало. I. Положить $y^1 = (\underbrace{0, 0, \ldots, 0}_{m_1})$.

II. Положить $l = 1$.

Основной цикл. III. Положить $y = y^l$.

IV. Решить задачу линейного программирования:

$$\text{найти } \arg\max_{x} \left(\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m_1} y_i \left(b_i^1 - \sum_{j=1}^{n} \alpha_{ij}^1 x_j\right)\right)$$

при ограничениях (4.27) и (4.28);
обозначить ее решение через $x^*(y^l)$, а через $z(y^l)$ — вектор оптимального решения двойственной задачи.

V. Вычислить вектор $h^l = (h_1^l, \ldots, h_{m_1}^l)$,

$$h_i^l = b_i^1 - \sum_{j=1}^{n} \alpha_{ij}^1 x_j^*(y^l), \quad i=1, \ldots, m_1.$$

VI. Вычислить шаговый множитель ρ_l.

VII. Вычислить вектор

$$y^{l+1} = \pi^+ (y^l - \rho_l h^l \| h^l \|^{-1}),$$

где π^+ — оператор, оставляющий неотрицательные координаты без изменений и обращающий отрицательные координаты в нуль.

VIII. Положить $l = l + 1$ и перейти к шагу III.

Теорема 3. *Если выполняются предположения 3 и шаговые множители удовлетворяют условиям*

$$\rho_l > 0, \quad l = 1, 2, \ldots; \quad \rho_l \underset{l \to \infty}{\to} 0; \quad \sum_{l=1}^{\infty} \rho_l = \infty,$$

то последовательность $\{\varphi^* (y^l)\}_{l=1}^{\infty}$ сходится к оптимальному значению функции $\varphi^* (y)$ на множестве $y \geqslant 0$, при этом пара векторов $(y^l, z(y^l))$ сходится к оптимальному решению двойственной задачи 3'.

Библиографические указания. Пункты 1, 2 написаны на основании работ [88, 115, 457], пункт 3 — на основании работы [394]. Дополнительные сведения о методах блочного программирования можно найти в работах [20, 52, 55, 458, 168—170, 166, 182, 211, 237, 258, 347, 353, 361, 495, 79].

4.6. Модификация симплекс-метода для решения задачи линейного программирования с двусторонними ограничениями

З а д а ч а 0. Найти $\arg\max\limits_{x \in X} \sum\limits_{j=1}^{n} c_j x_j$ для заданного вектора $c = (c_1, \ldots, c_n)$ и заданного множества ограничений

$$X \triangleq \{x \mid Ax = b, \ \alpha_j \leqslant x_j \leqslant \beta_j, \ \ j = 1, \ldots, n; \ \ x \in \mathbf{R}^n\},$$

где A — $m \times n$-матрица; b — n-мерный вектор; α_j, β_j, $j \in [1:n]$ — некоторые числа.

Предположения 0. (*i*) — ранг матрицы $A \triangleq (\alpha_{ij})_{i=1,\ldots,m}^{j=1,\ldots,n} = (a^1, \ldots, a^n)$ равен m; (*ii*) — $n > m$; (*iii*) — множество ограничений X непусто.

Определение 1. Допустимое решение $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ задачи 0 называется опорным, если система векторов a^j, соответствующих компонентам $\bar{x}_j$, для которых

$$\alpha_j < \bar{x}_j < \beta_j, \tag{4.29}$$

линейно-независима.

Определение 2. Система m линейно-независимых векторов, содержащая совокупность всех векторов, которым соответствуют компоненты $\bar{x}_j$ опорного решения $\bar{x}$, удовлетворяющие (4.29), называется базисом опорного решения $\bar{x}$.

Определение 3. Опорное решение $\bar{x}$ задачи 0 называется невырожденным, если m его компонент удовлетворяют неравенствам (4.29). Задача 0 называется невырожденной, если все ее опорные решения невырожденные.

Ниже приводится модифицированный симплекс-метод решения задачи линейного программирования с двусторонними ограничениями применительно к виду задачи 0. Для начала работы алгоритма требуется иметь опорное решение и его базис. Метод приводит к решению задачи 0 (либо устанавливает неограниченность линейной формы на множестве X) за число итераций, не превышающее $N = C_n^m 2^{n-m}$ (эта оценка справедлива для невырожденной задачи 0).

Если алгоритм 1 применять для решения вырожденной задачи 0, то (теоретически) возможно зацикливание (т. е. возврат к уже пройденному базису). Если при решении задачи 0 алгоритмом 1 возникает зацикливание, следует применять процедуру выбора вектора a^{i_r}, выводимого из базиса, исключающую зацикливание. В начальной стадии алгоритма требуется вычислять матрицу, обратную к исходной базисной.

1. Основной алгоритм

Алгоритм 1

Начало. I. Найти опорное решение $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ задачи 0 и его базис $a^{i_1}, \ldots, a^{i_m}$ (точнее множество $\mathcal{I}_{\bar{x}}$, состоящее из индексов $i_1, \ldots, i_m$, где $i_k \in [1 : n]$ при $k = 1, \ldots, m$).

II. Положить $z_{k0} = \bar{x}_{i_k}$, $k = 1, \ldots, m$.

Числа z_{k0}, $k = 1, \ldots, m$, являются коэффициентами разложения вектора $a^0 = b - \sum\limits_{j \in \mathcal{I}_{\bar{x}}} \bar{x}_j a^j$ по исходному базису (здесь $\mathcal{I}_{\bar{x}}$ — множество, состоящее из базисных индексов $i_1, \ldots, i_m$).

III. Вычислить матрицу B^{-1}, обратную к исходной базисной матрице B, состоящей из векторов $a^{i_1}, \ldots, a^{i_m}$.

IV. Разложить по базису $a^{i_1}, \ldots, a^{i_m}$ векторы a^j, $j = 1, \ldots, n$ (т. е. найти числа z_{kj} такие, что $a^j = \sum\limits_{k=1}^{m} z_{kj} a^{i_k}$, $j = 1, \ldots, n$) по правилу

$$(z_{1j}, \ldots, z_{mj})^T = B^{-1} (\alpha_{1j}, \ldots, \alpha_{mj})^T, \quad j = 1, \ldots, n.$$

Заранее известно разложение базисных векторов a^{i_k}, $k = 1, \ldots, m$: $z_{li_k} = 0$ при $k \neq l$ и $z_{li_k} = 1$ при $k = l$, $k = 1, \ldots, m$; $l = 1, \ldots, m$. Заранее известно, что для базисных векторов $\Delta_j = 0$, т. е. при $j \in \mathcal{I}_{\bar{x}}$ $\Delta_j = 0$.

V. Вычислить оценки

$$\Delta_j = \begin{cases} y_j, & \text{если } \alpha_j \leqslant x_j < \beta_j; \\ -y_j, & \text{если } x_j = \beta_j, \end{cases}$$

где $y_j = \sum_{k=1}^{m} c_{i_k} z_{kj} - c_j, \quad j = 1, \ldots, n.$

Основной цикл. VI. Если все $\Delta_j \geqslant 0$ ($j = 1, \ldots, n$), то положить $x^* = \bar{x}$ и прекратить вычисления (в этом случае опорное решение $\bar{x}$ является оптимальным решением задачи 0); иначе (т. е. если при некотором $j \in [1:n]$ выполняется $\Delta_j < 0$) перейти к шагу VII.

VII. Если существует индекс $j \in [1:n] \setminus \mathcal{J}_{\bar{x}}$, для которого $\Delta_j < 0$ и выполняется одно из условий (i) — при $\bar{x}_j = \alpha_j$ для $z_{kj} > 0$ $\alpha_{i_k} = -\infty$, для $z_{kj} < 0$ $\beta_{i_k} = \infty$; (ii) — при $\bar{x}_j = \beta_j$ для $z_{kj} > 0$ $\beta_{i_k} = \infty$, для $z_{kj} < 0$ $\alpha_{i_k} = -\infty$, то прекратить вычисления (в этом случае функция цели задачи 0 неограничена на допустимом множестве X); иначе перейти к шагу VII.

VIII. Найти индекс $s \in [1:n]$ такой, что $\Delta_s = \min_{j \in [1:n]} \Delta_j$ и положить $a^{i_{m+1}} = a^s$ (т. е. получить расширенный базис $a^{i_1}, \ldots, a^{i_m}, a^{i_{m+1}}$).

Индекс s принадлежит множеству $[1:n] \setminus \mathcal{J}_{\bar{x}}$ и определяет вектор a^s, который может включаться в старый базис. В качестве s можно брать любой индекс, для которого $\Delta_s < 0$.

IX. Положить $z_{m+1,s} = -1$, $z_{m+1,0} = \bar{x}_s$.

X. Если $\bar{x}_s = \alpha_s$, то положить $\nu = 0$ и перейти к шагу XI; если $\bar{x}_s = \beta_s$, то положить $\nu = 1$ и перейти к шагу XIII.

XI. Вычислить величины γ_k при таких $k \in [1:(m+1)]$, что $z_{ks} \neq 0$:

$$\gamma_k = \begin{cases} \alpha_{i_k} & \text{при } z_{ks} > 0; \\ \beta_{i_k} & \text{при } z_{ks} < 0. \end{cases}$$

XII. Вычислить параметр θ_0 преобразования опорного решения $\bar{x}$

$$\vartheta_0 = \min_{\substack{z_{ks} \neq 0 \\ 1 \leqslant k \leqslant m+1}} \frac{z_{k0} - \gamma_k}{z_{ks}},$$

обозначить через r индекс такой, что $\theta_0 = \dfrac{z_{r0} - \gamma_r}{z_{rs}}$ и перейти к шагу XV.

В невырожденном случае индекс r определяется однозначно. В вырожденном случае в качестве r выбирается наименьший индекс, на котором достигается минимум. Если в вырожденном случае возникает зацикливание, то следует использовать правило, выводящее из цикла.

XIII. Вычислить величины γ_k при $k \in [1 : (m+1)]$, для которых $z_{ks} \neq 0$:

$$\gamma_k = \begin{cases} \alpha_{i_k}, & \text{если } z_{ks} < 0; \\ \beta_{i_k}, & \text{если } z_{ks} > 0. \end{cases}$$

XIV. Вычислить параметр θ_0 преобразования опорного решения $\bar{x}$ по правилу

$$\theta_0 = \min_{\substack{z_{ks} \neq 0 \\ 1 \leqslant k \leqslant m+1}} \frac{\gamma_k - z_{k0}}{z_{ks}},$$

обозначить через r индекс, такой, что $\theta_0 = \dfrac{\gamma_r - z_{r0}}{z_{rs}}$ и перейти к шагу XV (см. шаг XII).

XV. Положить $x' = \bar{x}$; $z'_{kj} = z_{kj}$, $k = 1, \ldots, m$, $j = 1, \ldots, n$; $y'_j = y_j$, $j = 1, \ldots, n$; $\Delta'_j = \Delta_j$, $j = 1, \ldots, n$, и перейти к шагу XVI.

XVI. Вычислить новое опорное решение $\bar{x} = (\bar{x}_1, \ldots, \bar{x}_n)$ по правилу

$$\bar{x}_l = \begin{cases} x'_{i_k} - (-1)^{\nu} \theta_0 z'_{ks}, & \text{если } l = i_k, \quad k = 1, \ldots, m; \\ x'_s + (-1)^{\nu} \theta_0, & \text{если } l = s; \\ x'_l, & \text{если } l \notin \mathcal{J}_{\bar{x}}, \quad l \neq s. \end{cases}$$

XVII. Положить $i_r = s$, т. е. заменить в расширенном базисе $a^{i_1}, \ldots, a^{i_m}, a^{i_{m+1}}$ значение вектора a^{i_r} значением вектора a^s и перейти к новому базису $a^{i_1}, \ldots, a^{i_m}$, который получается из первых m векторов расширенного базиса (при переходе к новому базису старый базис не меняется, если $r = m + 1$ и в старом базисе меняется значение одного вектора a^{i_r}, если $r < m + 1$).

XVIII. Найти коэффициенты z_{kj}, $k = 1, \ldots, m$; $j = 1, \ldots, n$, разложения векторов $a^1, \ldots, a^n$ по векторам нового базиса:

а) если $r = m + 1$ (если базис не изменился), то положить

$$z_{kj} = z'_{kj}, \quad k = 1, \ldots, m; \quad j = 1, \ldots, n;$$

б) если $r < m + 1$ (т. е. если базис изменился), то вычислить z_{kj} по основным формулам

$$z_{rj} = z'_{rj}/z'_{rs}, \quad j = 1, \ldots, n;$$

$$z_{kj} = z'_{kj} - (z'_{rj}/z'_{rs})\, z'_{ks},$$

$$j = 1, \ldots, n; \quad k = 1, \ldots, r-1, r+1, \ldots, m.$$

XIX. Положить $z_{k0} = \bar{x}_{i_k}$, $k = 1, \ldots, m$.

XX. Вычислить оценки Δ_j, $j = 1, \ldots, n$, отвечающие новому базису:

а) если $r = m + 1$ (базис не изменился), то положить

$$\Delta_j = \begin{cases} \Delta_j' & \text{при } j \neq s; \\ -\Delta_j' & \text{при } j = s; \end{cases}$$

б) если $r < m + 1$ (базис изменился), то вычислить величины y_j, $j = 1, \ldots, n$ по основным формулам

$$y_j = y_j' - (z_{rj}'/z_{rs}')\, y_s', \quad j = 1, \ldots, n,$$

и положить

$$\Delta_j = (-1)^{\nu_j} y_j, \quad j = 1, \ldots, n,$$

где $\nu_j = 0$, если $\bar{x}_j = \alpha_j$, и $\nu_j = 1$, если $\bar{x}_j = \beta_j$.

Заранее известно, что $\Delta_j = 0$, если $j \in \mathcal{J}_{\bar{x}}$. Поэтому вычисления Δ_j следует проводить для $j \notin \mathcal{J}_{\bar{x}}$, т. е. когда или $\bar{x}_j = \alpha_j$, или $\bar{x}_j = \beta_j$.

XXI. Перейти к шагу VI.

Теорема 1. *Если выполнены предположения 0, то в невырожденном случае процесс решения задачи 0 алгоритмом 1 через конечное число итераций закончится либо на шаге VI (находится оптимальное решение задачи 0), либо на шаге VII (устанавливается, что функция цели задачи 0 неограничена на допустимом множестве X).*

Алгоритм 1 может применяться также для решения вырожденных задач линейного программирования с двусторонними ограничениями. Только при возникновении цикла следует использовать специальное правило вывода вектора из базиса, исключающее зацикливание.

2. Правило выбора индекса r (определяющего вектор a^{i_r}, выводимый из базиса) для предотвращения зацикливания

Если при использовании алгоритма 1 для решения задачи линейного программирования в вырожденном случае возникло зацикливание, то на шаге XII или XIV алгоритма следует применить следующую процедуру выбора индекса r, гарантирующую от зацикливания.

Алгоритм 2

Начало. I. Вычислить линейно-независимую систему векторов w^k, $k = 1, \ldots, m$,

$$w^k = \begin{cases} a^{i_k} & \text{при } z_{k0} < \beta_{i_k}; \\ -a^{i_k} & \text{при } z_{k0} = \beta_{i_k}. \end{cases}$$

II. Вычислить y_{kj}, $k = 1, \ldots, m$; $j = 1, \ldots, m$,— коэффициенты разложения векторов w^k, $k = 1, \ldots, m$, по базису $a^{i_1}, \ldots, a^{i_m}$, т. е.

такие, что

$$w^j = \sum_{k=1}^{m} y_{kj} a^{l_k}, \quad j = 1, \ldots, m.$$

III. Положить $y_{m+1,j} = 0$, $j = 1, 2, \ldots, m$.

Основной цикл. IV. Если $\bar{x}_s = \alpha_s$, то перейти к шагу V; если $\bar{x}_s = \beta_s$, то перейти к шагу X;

V. Найти множество G_0, состоящее из тех индексов k, на которых

$$\theta_0 = \min_{\substack{z_{ks} \neq 0 \\ 1 \leqslant k \leqslant m+1}} \frac{z_{k0} - \gamma_k}{z_{ks}}.$$

VI. Положить $\mu = 0$.

VII. Если G_μ состоит из одного элемента, то принять r равным этому элементу и прекратить вычисления; иначе (если G_μ состоит более чем из одного элемента) перейти к шагу VIII.

VIII. Найти множество $G_{\mu+1}$, состоящее из индексов $k \in G_\mu$, на которых

$$\min_{k \in G_\mu} (y_{k,\mu+1}/z_{ks}).$$

IX. Положить $\mu = \mu + 1$ и перейти к шагу VII.

X. Найти множество G_0, состоящее из индексов k, на которых

$$\theta_0 = \min_{\substack{z_{ks} \neq 0 \\ 1 \leqslant k \leqslant m+1}} \frac{\gamma_k - z_{k0}}{z_{ks}}.$$

XI. Положить $\mu = 0$.

XII. Если G_μ состоит из одного элемента, то принять r равным этому элементу и прекратить вычисления; иначе (если G_μ состоит более чем из одного элемента) перейти к шагу XIII.

XIII. Найти множество $G_{\mu+1}$, состоящее из индексов $k \in G_\mu$, на которых достигается

$$\max_{k \in G_\mu} (y_{k,\mu+1}/z_{ks}).$$

XIV. Положить $\mu = \mu + 1$ и перейти к шагу XII.

Теорема 2. *Алгоритм 2 генерирует конечную последовательность множеств $G_0, G_1, \ldots, G_t$, $t \leqslant m$, такую, что G_t содержит единственный элемент, который принимается за r.*

Библиографические указания. При написании параграфа использовались работы [414, 415, 416].

4.7. Модификация симплекс-метода для решения общей задачи линейного программирования

Задача 1. Найти $\arg\max_x \sum_{j=1}^{n} c_j x_j$ для заданного вектора $c = (c_1, c_2, \ldots, c_n)$ при ограничениях

$$\sum_{j=1}^{n} \alpha_{ij} x_j = b_i, \quad i = 1, \ldots, p; \tag{4.30}$$

$$\sum_{j=1}^{n} \alpha_{ij} x_j \leqslant b_i, \quad i = p+1, \ldots, m; \tag{4.31}$$

$$x_j \geqslant 0, \quad j = q+1, \ldots, n. \tag{4.32}$$

Предположения 1. (i) — уравнения (4.30) линейно-независимы; (ii) — векторы $a^j = (\alpha_{1j}, \alpha_{2j}, \ldots, \alpha_{mj})^T$, $j = 1, \ldots, q$ — линейно-независимы; (iii) — задача 1 имеет допустимое решение.

Определение 1. Допустимое решение $x = (x_1, \ldots, x_n)$ задачи 1 называется опорным, если оно порождает неособенную квадратную подматрицу

$$A_x \triangleq \begin{pmatrix} \alpha_{i_1 j_1} & \alpha_{i_1 j_2} & \ldots & \alpha_{i_1 j_s} \\ \alpha_{i_2 j_1} & \alpha_{i_2 j_2} & \ldots & \alpha_{i_2 j_s} \\ \cdot & \cdot & \cdot & \cdot \\ \alpha_{i_s j_1} & \alpha_{i_s j_2} & \ldots & \alpha_{i_s j_s} \end{pmatrix}$$

матрицы $A \triangleq (\alpha_{ij})_{i=1,\ldots,m}^{j=1,\ldots,n}$ такую, что:

а) столбцы матрицы A_x содержат все столбцы матрицы A, для которых компонента x_j положительна в x или не ограничена требованием неотрицательности, $j_l = l$ для $l = 1, 2, \ldots, q$;

б) строки матрицы A_x образованы условиями (4.30) и некоторыми из условий (4.31), причем все такие ограничения-неравенства удовлетворяются допустимым решением x как строгие равенства, $i_k = k$ для $k = 1, \ldots, p$.

Подматрица A_x, порождаемая опорным решением x, называется его базисом.

Определение 2. Опорное решение x задачи 1 с базисом A_x называется невырожденным, если

$$x_{j_l} > 0 \text{ при } l = q+1, q+2, \ldots, s;$$

$$\sum_{j=1}^{n} \alpha_{ij} x_j < b_i \text{ при } i \neq i_k \ (k = 1, \ldots, s).$$

С одной стороны, задача 1 может быть решена методами, приведенными выше, если предварительно свести ее к каноническому виду с $n + m - (p + q)$ неотрицательными переменными и $m - q$ условиями равенствами, с другой — существует ряд классов задач линейного программирования (например, двойственные задачи при

$m < n - m$), которые выгоднее решать в их естественной записи. Ниже приводится модификация симплекс-метода для решения задачи 1 в естественной записи. В начальной стадии алгоритма требуется иметь опорное решение и его базис, а также необходимо вычислять матрицу, обратную к исходной базисной. На каждой итерации алгоритма опорное решение проверяется на оптимальность; если оно не оптимально, то осуществляется переход к новому опорному решению и новому базису (лучшему от предыдущего, т. е. дающему большее значение целевой функции). В невырожденном случае за конечное число итераций находится оптимальное решение x^*, либо устанавливается неограниченность линейной формы на допустимом множестве решений. В вырожденном случае теоретически может возникнуть зацикливание, тогда необходимо применять специальное правило, гарантирующее от зацикливания.

Алгоритм 1

Начало. I. Найти исходное опорное решение $x = (x_1, \ldots, x_n)$ задачи 1 и его базис A_x, образованный элементами, стоящими на пересечении строк и столбцов матрицы A с номерами $i_1, \ldots, i_s$ и $j_1, \ldots, j_s$, соответственно.

II. Вычислить матрицу $A_x^{-1} = (\beta_{lk})_{l=1,\ldots,s}^{k=1,\ldots,s}$, обратную к матрице A_x.

Основной цикл. III. Вычислить числа

$$\lambda_k = \sum_{l=1}^{s} c_{j_l}\beta_{lk}, \quad k = 1, \ldots, s.$$

IV. Вычислить величины

$$\Delta_j = \sum_{k=1}^{s} \alpha_{i_k j}\lambda_k - c_j, \quad j = 1, \ldots, n.$$

V. Если выполняются неравенства

$$\lambda_k \geqslant 0, \quad k = p+1, \ldots, s;$$
$$\Delta_j \geqslant 0, \quad j = 1, \ldots, n,$$

то положить $x^* = x$ и прекратить вычисления (в этом случае опорное решение x является оптимальным решением задачи 1); иначе положить $\overline{A}_x = A_x$ (для этого надо положить $\overline{j}_l = j_l$, $l = 1, \ldots, s$; $\overline{i}_k = i_k$, $k = 1, \ldots, s$); $\overline{x} = x$; $\overline{\beta}_{lk} = \beta_{lk}$; $l = 1, \ldots, s$; $k = 1, \ldots, s$ и перейти к шагу VI.

VI. Среди величин λ_k, $k = p + 1, \ldots, s$, и Δ_j, $j = 1, \ldots, n$, найти наименьшую. Если наименьшей величиной оказалось Δ_r, $r \in [1 : n]$, то перейти к шагу VII, а если λ_t, $t \in [(p + 1) : s]$, то перейти к шагу XXI.

VII. Вычислить величины

$$z_{lr} = \sum_{k=1}^{s} \beta_{lk}\alpha_{i_k r}, \quad l = 1, \ldots, s.$$

VIII. Вычислить параметр

$$\theta_0' = \begin{cases} \min\limits_{\substack{z_{lr}>0 \\ l>q}} (x_{j_l}/z_{lr}), & \text{если существует } z_{lr} > 0, \ l > q; \\ \infty, \text{ если } z_{lr} \leqslant 0 & \text{для } l = q+1, \ldots, s. \end{cases}$$

IX. Вычислить параметр θ_0'' по формуле

$$\theta_0'' = \begin{cases} \min\limits_{\delta_i<0} (-\Delta^{(i)}/\delta_i), & \text{если существует } \delta_i < 0, \ i \in [1:m]; \\ \infty, & \text{если } \delta_i \geqslant 0, \quad i = 1, \ldots, m, \end{cases} \tag{4.33}$$

где

$$\Delta^{(i)} = b_i - \sum_{j=1}^{n} \alpha_{ij} x_j, \quad i = 1, \ldots, m;$$

$$\delta_i = \sum_{l=1}^{s} \alpha_{ij_l} z_{lr} - \alpha_{ir}, \quad i = 1, \ldots, m.$$

X. Вычислить параметр θ_0 по формуле

$$\theta_0 = \min \{\theta_0', \theta_0''\}.$$

Если $\theta_0 = \infty$, то прекратить вычисления (в этом случае целевая функция задачи 1 неограничена на допустимом множестве); если $\theta_0 < \infty$, то перейти к шагу XI.

XI. Вычислить новое опорное решение $x = (x_1, \ldots, x_n)$ по формуле

$$x = \bar{x} - \theta_0 h,$$

где $h = (h_1, \ldots, h_n)$ — вектор, определяемый по правилу

$$h_j = \begin{cases} z_{lr}, & \text{если } j = j_l, \ l = 1, \ldots, s; \\ -1, & \text{если } j = r; \\ 0, & \text{в остальных случаях.} \end{cases}$$

XII. Если $\theta_0 = \theta_0'$, то перейти к шагу XIII, если $\theta_0 = \theta_0''$, то перейти к шагу XVII.

XIII. Вычислить индекс $\nu \in [(q+1) : s]$ такой, что

$$\theta_0' = x_{j_\nu}/z_{\nu r}, \ z_{\nu r} > 0. \tag{4.34}$$

XIV. Перейти к новому базису A_x, который получается из старого базиса $\bar{A}_x$ путем замены его ν-столбца столбцом

$$(\alpha_{i_1 r}, \alpha_{i_2 r}, \ldots, \alpha_{i_s r})^T \text{ (т. е. положить } j_\nu = r).$$

XV. Вычислить матрицу $A_x^{-1} = (\beta_{lk})_{l=1,\dots,s}^{k=1,\dots,s}$, обратную новой базисной матрице по правилам

$$\beta_{lk} = \begin{cases} \bar\beta_{lk} - \bar\beta_{\nu k} z_{lr} \frac{1}{z_{\nu r}}, & l \neq \nu; \\ \bar\beta_{\nu k} \frac{1}{z_{\nu r}}, & l = \nu, \end{cases}$$

где $l = 1, \dots, s$ и $k = 1, \dots, s$.

XVI. Перейти к шагу III.

XVII. Вычислить индекс $\mu \in [1 : m]$ такой, что $\theta_0'' = -\Delta^{(\mu)}/\delta_\mu$, $\delta_\mu < 0$; $\mu \neq i_k$, $k = 1, \dots, s$.

XVIII. Перейти к новому базису A_x, который получается путем окаймления базиса $\bar A_x$ строкой $(\alpha_{\mu j_1}, \alpha_{\mu j_2}, \dots, \alpha_{\mu j_s}, \alpha_{\mu r})$ и столбцом $(\alpha_{i_1 r}, \alpha_{i_2 r}, \dots, \alpha_{i_s r}, \alpha_{\mu r})^T$; положить $i_{s+1} = \mu$, $j_{s+1} = r$ и перейти к шагу XIX.

XIX. Вычислить матрицу $A_x^{-1} = (\beta_{lk})_{l=1,\dots,s+1}^{k=1,\dots,s+1}$, обратную новой базисной матрице по правилам

$$\beta_{lk} = \begin{cases} \bar\beta_{lk} + \hat\delta_k z_{lr}/\alpha_0 & \text{для } l, k = 1, \dots, s; \\ -\hat\delta_k/\alpha_0 & \text{для } l = s+1, k = 1, \dots, s; \\ -z_{lr}/\alpha_0 & \text{для } l = 1, \dots, s, k = s+1; \\ 1/\alpha_0 & \text{для } l = k = s+1, \end{cases}$$

где $\hat\delta_k$ и α_0 вычисляются, соответственно, по формулам

$$\hat\delta_k = \sum_{\gamma=1}^{s} \alpha_{\mu j_\gamma} \bar\beta_{\gamma k}, \quad k = 1, \dots, s;$$

$$\alpha_0 = \alpha_{\mu r} - \sum_{\gamma=1}^{s} \alpha_{\mu j_\gamma} z_{\gamma r}.$$

XX. Положить $s = s + 1$ и перейти к шагу III.

XXI. Найти параметр θ_0'' из условия

$$\theta_0'' = \begin{cases} \infty, & \text{если } \hat\delta^{(i)} \geqslant 0, \quad i = 1, \dots, m; \\ \min\limits_{\substack{\hat\delta^{(i)} < 0 \\ 1 \leqslant i \leqslant m}} (-\Delta^{(i)}/\hat\delta^{(i)}), & \text{если существует } \hat\delta^{(i)} < 0; \end{cases} \quad (4.35)$$

$$\Delta^{(i)} = b_i - \sum_{j=1}^{n} \alpha_{ij} x_j, \quad i = 1, \dots, m;$$

$$\hat\delta^{(i)} = \sum_{\gamma=1}^{s} \alpha_{i j_\gamma} \beta_{\gamma t}, \quad i = 1, \dots, m.$$

XXII. Найти параметр θ_0' из условия

$$\theta_0' = \begin{cases} \min\limits_{\substack{\beta_{\gamma t}>0\\ \gamma>q}} (x_{j_\gamma}/\beta_{\gamma t}), & \text{если существует } \beta_{\gamma t}>0,\ \gamma>q, \\ \infty, & \text{если } \beta_{\gamma t}\leqslant 0,\ \gamma=q+1,\ \ldots,\ s. \end{cases} \quad (4.36)$$

XXIII. Вычислить параметр θ_0 по формуле

$$\theta_0 = \min\{\theta_0',\ \theta_0''\}.$$

Если $\theta_0 = \infty$, то прекратить вычисления (в этом случае целевая функция задачи 1 неограничена на допустимом множестве); если $\theta_0 < \infty$, то перейти к шагу XXIV.

XXIV. Вычислить новое опорное решение $x = (x_1,\ ...,\ x_n)$ по формуле

$$x = \bar{x} - \theta_0 h,$$

где $h = (h_1,\ ...,\ h_n)$ — вектор, определяемый по правилу

$$h_j = \begin{cases} \beta_{lt}, & \text{если } j = j_l,\ l = 1,\ \ldots,\ s, \\ 0, & \text{в остальных случаях.} \end{cases}$$

XXV. Если $\theta_0 = \theta_0'$, то перейти к шагу XXVI, если $\theta_0 = \theta_0''$, то перейти к шагу XXX.

XXVI. Вычислить индекс $\nu \in [(q+1) \mathbin{:} s]$ такой, что

$$\theta_0' = (x_{j_\nu}/\beta_{\nu t}),\quad \beta_{\nu t} > 0.$$

Если задача 1 невырожденная, то индекс ν определяется однозначно. В вырожденном случае можно выбрать любой индекс ν, на котором достигается минимум в (4.36).

XXVII. Перейти к новому базису A_x, который получается из старого базиса $\bar{A}_x$ путем удаления его t-й строки и ν-го столбца, (т. е. положить $j_1 = \bar{j}_1,\ \ldots,\ j_{\nu-1} = \bar{j}_{\nu-1},\ j_\nu = \bar{j}_{\nu+1},\ \ldots,\ j_{s-1} = \bar{j}_s$; $i_1 = \bar{i}_1,\ \ldots,\ i_{t-1} = \bar{i}_{t-1},\ i_t = \bar{i}_{t+1},\ \ldots,\ i_{s-1} = \bar{i}_s$, где $\bar{j}_l,\ \bar{i}_k,\ l = 1,\ \ldots,\ s,\ k = 1,\ \ldots,\ s$ определяют старый базис $\bar{A}_x$).

XXVIII. Вычислить матрицу $A_x^{-1} = (\beta_{lk})_{l=1,\ldots,s-1}^{k=1,\ldots,s-1}$, обратную к новой базисной матрице по правилам

$$\beta_{lk} = \begin{cases} \bar{\beta}_{lk} - (\bar{\beta}_{\nu k}\bar{\beta}_{lt}/\bar{\beta}_{\nu t}), & \text{если } l<\nu,\ k<t; \\ \bar{\beta}_{l+1,k} - (\bar{\beta}_{\nu k}\bar{\beta}_{l+1,t}/\bar{\beta}_{\nu t}), & \text{если } l\geqslant\nu,\ k<t; \\ \bar{\beta}_{l,k+1} - (\bar{\beta}_{\nu,k+1}\bar{\beta}_{lt}/\bar{\beta}_{\nu t}), & \text{если } l<\nu,\ k\geqslant t; \\ \bar{\beta}_{l+1,k+1} - (\bar{\beta}_{\nu,k+1}\bar{\beta}_{l+1,t}/\bar{\beta}_{\nu t}), & \text{если } l\geqslant\nu,\ k\geqslant t, \\ l = 1,\ \ldots,\ s-1,\ k = 1,\ \ldots,\ s-1. \end{cases}$$

XXIX. Положить $s = s - 1$ и перейти к шагу III.

XXX. Вычислить индекс $\mu \in [1 \mathbin{:} m]$ такой, что

$$\theta_0'' = -\Delta^{(\mu)}/\hat{\delta}^{(\mu)},\ \hat{\delta}^{(\mu)} < 0;\quad \mu \neq i_k,\quad k = 1,\ \ldots,\ s.$$

Если задача 1 невырожденная, то индекс μ определяется однозначно. В случае вырожденной задачи можно выбрать любой индекс μ, на котором достигается минимум в (4.35).

XXXI. Вычислить величины

$$\hat{\delta}_k = \sum_{l=1}^{s} \alpha_{\mu j_l} \beta_{lk}, \quad k = 1, \ldots, s.$$

XXXII. Перейти к новому базису A_x, который получается из старого базиса $\bar{A}_x$ путем замены его t-й строки строкой $(\alpha_{\mu j_1}, \alpha_{\mu j_2}, \ldots$ $\ldots, \alpha_{\mu j_s})$ (т. е. положить $i_t = \mu$).

XXXIII. Вычислить матрицу $A_x^{-1} = (\beta_{lk})_{l=1,\ldots,s}^{k=1,\ldots,s}$, обратную новой базисной матрице по правилам

$$\beta_{lk} = \begin{cases} \bar{\beta}_{lk} - (\bar{\beta}_{lt}\hat{\delta}_k/\hat{\delta}^{(\mu)}), & k \neq t; \\ \bar{\beta}_{lt}/\hat{\delta}^{(\mu)}, & k = t, \end{cases} \quad l, k = 1, \ldots, s.$$

XXXIV. Перейти к шагу III.

Теорема 1. *Если выполнены предположения 1, то в невырожденном случае процесс решения задачи 1 алгоритмом 1 через конечное число итераций закончится либо на шаге V (находится оптимальное решение x^* задачи 1), либо на одном из шагов X или XXIII (устанавливается, что функция цели задачи 1 неограничена на допустимом множестве решений).*

Библиографические указания. Параграф написан на основании работ [414, 415, 416].

4.8. Итеративные методы

Итеративные методы представляют собой последовательность однообразных по процедуре выполнения итераций, которые приводят в пределе к оптимальному решению задачи. Их применение целесообразно в тех случаях, когда требуется достаточно быстро получить грубое решение задачи, а также при решении задач большого размера. Итеративные методы более помехоустойчивы, чем конечные методы линейного программирования.

1. Итеративный метод Петшиковского

Задача 1. Найти $\arg\max_x (c, x)$ для заданного вектора $c \in R^n$ при ограничениях

$$\begin{aligned} &\varphi_i(x) \triangleq (\bar{a}^i, x) - a_i^0 \geqslant 0, \quad i = 1, 2, \ldots, m; \\ &\varphi_{m+j}(x) \triangleq x_j = (e^j, x) \geqslant 0, \quad j = 1, 2, \ldots, n_1, \quad (n_1 \leqslant n), \end{aligned} \qquad (4.37)$$

где $\bar{a}^i$ — i-й вектор-строка матрицы A задачи 1.0; e^j — n-мерный единичный вектор с единицей на j-м месте.

Если положить $\bar{a}^{m+i} = e^i$, $a^0_{m+i} = 0$, $i = 1, 2, \ldots, n_1$, то ограничения (4.37) задачи 1 можно переписать в следующей эквивалентной форме:

$$\varphi_i(x) \triangleq (\bar{a}^i, x) - a_i^0 \geqslant 0, \quad i = 1, 2, \ldots, m + n_1. \qquad (4.38)$$

В отличие от рассмотренных метод Петшиковского не является конечным. Решение задачи 1 получается лишь в пределе.

В методе Петшиковского задача 1 с ограничениями (4.38) приводится к задаче безусловной максимизации выпуклой экстремальной функции

$$g_\mu(x) \triangleq (c, x) + \frac{\mu}{2} \sum_{i=1}^{m+n_1} \delta_i(x) [\varphi_i(x)]^2,$$

где

$$\delta_i(x) = \begin{cases} 0, & \text{если } \varphi_i(x) \geqslant 0, \\ -1, & \text{если } \varphi_i(x) < 0. \end{cases}$$

Для решения задачи безусловной оптимизации применяется градиентный метод, который при достаточно больших μ дает хорошее приближенное решение задачи 1.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать параметр $\mu > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить величины

$$\beta_i^k = \varphi_i(x^k), \quad i = 1, 2, \ldots, m + n_1.$$

V. Вычислить градиент функции g_μ в точке $x = x^k$

$$\nabla g_\mu(x^k) = c + \mu \sum_{i=1}^{m+n_1} \delta_i(x^k) \beta_i^k \bar{a}^i.$$

VI. Вычислить величины

$$\xi_i^k = (\bar{a}^i, \nabla g_\mu(x^k)), \quad i = 1, 2, \ldots, m + n_1.$$

VII. Для всех $i \in [1 : (m + n_1)]$, для которых $\xi_i^k \neq 0$, вычислить отношения $\theta_i^k = -\beta_i^k / \xi_i^k$ и расположить их в порядке возрастания

$$\theta_{i_1}^k \leqslant \theta_{i_2}^k \leqslant \cdots \leqslant \theta_{i_s}^k.$$

VIII. Определить функцию

$$\zeta(\theta) \triangleq \frac{\partial}{\partial \theta} g_\mu(x^k + \theta \nabla g_\mu(x^k)) = (c, \nabla g_\mu(x^k)) +$$

$$+ \mu \sum_{i=1}^{m+n_1} \delta_i(x^k + \theta \nabla g_\mu(x^k)) (\beta_i^k + \theta \xi_i^k) \xi_i^k.$$

IX. Найти индекс $l \in [1 : s]$, при котором выполняются неравенства

$$\zeta(\theta_{i_l}^k) > 0; \quad \zeta(\theta_{i_{l+1}}^k) \leqslant 0.$$

X. Вычислить шаговый множитель

$$\rho_k = [\zeta(\theta_{i_l}^k)\,\theta_{i_{l+1}}^k - \zeta(\theta_{i_{l+1}}^k)\,\theta_{i_l}^k]/[\zeta(\theta_{i_l}^k) - \zeta(\theta_{i_{l+1}}^k)].$$

XI. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k \nabla g_\mu(x^k).$$

XII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, при любом* $x^0 \in \boldsymbol{R}^n$ *и* $\mu > 0$ *обладает тем свойством, что*

$$\lim_{k\to\infty} d_1(x^k, X_\mu) = 0,$$

где X_μ *— множество, на котором достигается максимум функции* g_μ; $d_1(x, Y)$ *— расстояние от точки* x *к множеству* Y.

Теорема 1'. *Для любого* $\varepsilon > 0$ *существует такое* $\mu_0 > 0$, *что при любом* $\mu > \mu_0$ *неравенство*

$$d_1(x, X^*) < \varepsilon$$

выполняется для любого $x \in X_\mu$,
где X^* *— множество решений задачи 1.*

Из теорем 1 и 1' следует, что при достаточно больших значениях k и μ точка x^k, полученная с помощью алгоритма 1, достаточно хорошо приближает решение задачи 1.

2. Итеративный метод, использующий модифицированную функцию Лагранжа

З а д а ч а 2. Найти $\arg\min_{x\in X}(c, x)$ для заданного вектора $c \in \boldsymbol{R}^n$ и заданного множества

$$X \triangleq \{x \mid Ax = a^0, \quad x \geqslant 0, \quad x \in \boldsymbol{R}^n\},$$

где $a^0 \in \boldsymbol{R}^m$; A — матрица размера $m \times n$.

Предположение 2. Множество X^* решений задачи 2 непусто.

Приводимый ниже алгоритм основан на построении модифицированной функции Лагранжа и требует на каждой итерации решать задачу минимизации квадратичной функции. Этот метод дает решение задачи 2 за конечное число итераций k_1, зависящее от величины коэффициентов штрафа и от выбора начального приближения двойственной переменной (при этом предполагается, что задача минимизации квадратичной функции решается за конечное число итераций).

Алгоритм 2

Н а ч а л о. I. Выбрать произвольное начальное приближение двойственной переменной $y^0 \in \boldsymbol{R}^m$.

II. Выбрать произвольную последовательность штрафных коэффициентов $\{\alpha_k\}_{k=0}^{\infty}$, $\alpha_k > 0$, $k = 0, 1, \ldots$.

III. Положить $k = 0$.

Основной цикл. IV. Определить модифицированную функцию Лагранжа

$$\psi(x, y, \alpha_k) \triangleq (c, x) + (y, a^0 - Ax) + \frac{\alpha_k}{2} \|Ax - a^0\|^2.$$

V. Вычислить приближение $x^k \geqslant 0$ основной переменной из условия

$$\psi(x^k, y^k, \alpha_k) = \min_{x \geqslant 0} \psi(x, y^k, \alpha_k).$$

VI. Вычислить следующее приближение двойственной переменной

$$y^{k+1} = y^k + \alpha_k (a^0 - Ax^k).$$

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Если выполняется предположение 2 и последовательность штрафных коэффициентов $\{\alpha_k\}_{k=0}^{\infty}$ такова, что $\alpha_k \geqslant \alpha > 0$, $k = 0, 1, \ldots$, то алгоритм 2 конечен, т. е. найдется такой номер k_1, что $x^{k_1} \in X^*$, $y^{k_1} \in Y^*$, где X^*, Y^*— множество седловых точек функции Лагранжа*

$$\varphi(x, y) \triangleq (c, x) + (y, a^0 - Ax).$$

Число k_1 тем меньше, чем меньше отношение $\rho(y^0, Y^*)/\alpha$. В частности, для всякого y^0 найдется $\bar{\alpha}$ такое, что при $\alpha_0 \geqslant \bar{\alpha}$ алгоритм 2 закончится за одну итерацию, т. е. будет $x^1 \in X^*$, $y^1 \in Y^*$ (здесь $\rho(x, Y) = \inf_{y \in Y} \|x - y\|$).

Замечание 2. Важной проблемой в алгоритме 2 является выбор последовательности штрафных коэффициентов α_k, $k = 0, 1, \ldots$.

Увеличение α_k выгодно с точки зрения уменьшения числа итераций. Однако при больших значениях α_k трудно решать задачу минимизации функции $\psi(x, y^k, \alpha_k)$. Поэтому приемлемое значение α_k следует выбирать в процессе вычислений.

3. Итеративный метод Федоренко

Задача 3. Найти $\arg\min_{y \in Y} \sum_{i=1}^{n} y_i h_i^0$ для заданного вектора $h^0 = (h_1^0, \ldots, h_n^0)$ и заданного множества

$$Y \triangleq \Big\{ y \mid b_j + \sum_{i=1}^{n} y_i h_i^j = 0, \ j = 1, \ldots, m;$$
$$\alpha_i^- \leqslant y_i \leqslant \alpha_i^+, \quad i = 1, \ldots, n; \quad y \in R^n \Big\}.$$

Если отобразить точки $y = (y_1, \ldots, y_n)$ пространства R^n в точки $x = (x_1, \ldots, x_{m+1})$ пространства R^{m+1} по правилу

$$x = \begin{pmatrix} 0 \\ b_1 \\ \vdots \\ b_m \end{pmatrix} + y_1 \begin{pmatrix} h_1^0 \\ h_1^1 \\ \vdots \\ h_1^m \end{pmatrix} + \cdots + y_n \begin{pmatrix} h_n^0 \\ h_n^1 \\ \vdots \\ h_n^m \end{pmatrix}, \qquad (4.39)$$

то образом n-мерного «параллелепипеда»

$$\sigma = \{y \mid \alpha_i^- \leqslant y_i \leqslant \alpha_i^+, \quad i = 1, \ldots, n\}$$

будет выпуклый многогранник τ в пространстве R^{m+1}. Тогда метод решения задачи 3 сводится к отысканию минимального числа λ, для которого $\lambda e \in \tau$, где $e = (1, 0, \ldots, 0) \in R^{m+1}$ (минимальное значение λ обозначим через λ^*).

Приводимый ниже алгоритм определяет за конечное число итераций приближенное решение $\tilde{x}$ вспомогательной задачи и ему соответствующее приближенное решение $\bar{y}$ задачи 3 (точность приближенного решения $\tilde{x}$ задается константой δ_0). Объем вычислений существенно зависит от выбора параметров алгоритма (о методике выбора этих параметров см. [363, 364, 365, 366, 367]). На каждой основной итерации алгоритма 3 используется заданное число «малых» итераций любого выбранного алгоритма квадратичного программирования.

Алгоритм 3

Начало. I. Выбрать вектор $g \in R^{m+1}$ такой, что $(g, e) = 1$ (здесь $e = (1, 0, \ldots, 0)$).

II. Положить

$$a^i = (h_i^0, h_i^1, \ldots, h_i^m), \quad i = 1, \ldots, n;$$
$$c = (0, b_1, \ldots, b_m).$$

III. Задать число $l > 0$ — количество «малых» итераций решения вспомогательной задачи квадратичного программирования (обычно $l \in [10, 20]$).

IV. Задать константы $\beta \in (0, 1)$, $\varepsilon > 0$,

$$d^* \, (0 < d^* < 1), \quad \eta^* > 0, \quad \delta_0 > 0, \quad w_0 > 0.$$

Основной цикл. V. Вычислить величины

$$\varphi_i = (a^i, g), \quad i = 1, \ldots, n.$$

VI. Вычислить величины

$$\hat{\alpha}_i = \begin{cases} \alpha_i^-, & \text{если } \varphi_i \geqslant 0, \\ \alpha_i^+, & \text{если } \varphi_i < 0, \quad i = 1, \ldots, n. \end{cases}$$

VII. Вычислить вектор

$$\hat{x} = c + \sum_{i=1}^{n} \hat{\alpha}_i a^i.$$

VIII. Вычислить значение

$$f_0(g) = (\hat{x},\ g). \qquad \textbf{(4.40)}$$

Отметим, что точка $\hat{x}$ — решение задачи

$$\min_{x \in \tau} (x,\ g) = f_0(g),$$

при этом $f_0(g)$ дает точную оценку снизу для $\lambda^* : f_0(g) \leqslant \lambda^*$.

IX. Найти подмножество индексов $\mathcal{J}_\varepsilon$ ($\mathcal{J}_\varepsilon \subset \{1, \ldots, n\}$) по правилу

$$i \in \mathcal{J}_\varepsilon, \text{ если } |\varphi_i| \leqslant \varepsilon \| a^i \|_G,$$

где норма $\| z \|_G$ определяется как расстояние точки z от прямой λe $(-\infty < \lambda < \infty)$, отсчитываемое в плоскости G, проходящей через z ортогонально вектору g, и вычисляется по правилу

$$\| z \|_G = \| z - (z,\ g)\, e \|. \qquad \textbf{(4.41)}$$

X. Вычислить точку

$$\bar{x} = c + \sum_{i \notin \mathcal{J}_\varepsilon} \hat{\alpha}_i a^i.$$

XI. Положить $\nu = 1$.

XII. Выбрать алгоритм $\hat{A}$ для решения следующей задачи квадратичного программирования:

$$\text{найти arg} \min_{\substack{\alpha_i^- \leqslant y_i \leqslant \alpha_i^+ \\ i \in \mathcal{J}_\varepsilon}} \| \bar{x} + \sum_{i \in \mathcal{J}_\varepsilon} y_i a^i \|_G, \qquad \textbf{(4.42)}$$

где норма $\| \cdot \|_G$ определяется по правилу (4.41).

XIII. Найти приближенное решение y_i^ν, $i \in \mathcal{J}_\varepsilon$, задачи (4.42), применяя l итераций алгоритма $\hat{A}$.

XIV. Вычислить вектор

$$\tilde{x} = \bar{x} + \sum_{i \in \mathcal{J}_\varepsilon} y_i^\nu a^i.$$

XV. Вычислить величину

$$\eta = |(\tilde{x},\ g) - f_0(g)| / |f_0(g)|.$$

Если $\eta > \eta^*$, то положить $\varepsilon = \beta\varepsilon$ и перейти к шагу IX; если $\eta \leqslant \eta^*$, то перейти к шагу XVI.

XVI. Если $\| \tilde{x} - (\tilde{x},\ g) e \| \leqslant \delta_0$, то прекратить вычисления (в этом случае точка $\tilde{x}$ принимается за решение вспомогательной

задачи, а вектор $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_n)$, где

$$\bar{y}_i = \begin{cases} y_i^\nu, & i \in \mathcal{I}_\varepsilon; \\ \hat{\alpha}_i, & i \notin \mathcal{I}_\varepsilon, \quad i = 1, \ldots, n, \end{cases}$$

принимается за приближенное решение задачи 3); иначе перейти к шагу XVII.

XVII. Вычислить вектор

$$v = (\tilde{x} - (\tilde{x},\ g)\,e)/\|\tilde{x} - (\tilde{x},\ g)\,e\|.$$

XVIII. Вычислить значения

$$\psi_i = (v,\ a^i) - \varphi_i(e,\ v), \quad i = 1, \ldots, n.$$

XIX. Вычислить значение

$$d = \frac{1}{\|\tilde{x} - (\tilde{x},\ g)\,e\|} \sum_{i \in \mathcal{I}_\varepsilon} \psi_i \times \begin{cases} y_i^\nu - \alpha_i^+, & \text{если } \psi_i \leqslant 0, \\ y_i^\nu - \alpha_i^-, & \text{если } \psi_i > 0. \end{cases}$$

XX. Если $d > d^*$ (в этом случае задача (4.42) не решена с требуемой точностью), то положить $\nu = \nu + 1$ и перейти к шагу XIII, принимая за начальное приближение алгоритма $\hat{A}$ приближенное решение $y_i^{\nu-1}$ задачи (4.42), полученное на $(\nu - 1)$-й «малой» итерации; если $d \leqslant d^*$, то перейти к шагу XXI для пересчета вектора g.

(«Малые» итерации порождаются шагами XI—XX алгоритма 3. Новый вектор $\tilde{g}$ определяется как

$$\tilde{g} = \tilde{g}(\rho) = (1 - \rho(e,\ v))\,g + \rho v,$$

где параметр $\rho = \arg\max_{\rho} \min_{x \in \tau} (x,\ \tilde{g}(\rho))$. На шагах XXI—XXX алгоритма методом «деления вилки» вычисляется нужное значение ρ за конечное число итераций.

Величина $(\tilde{x},\ g)$ является верхней оценкой для λ^*, точной лишь при $\|\tilde{x}\|_G = 0$).

XXI. Вычислить значение

$$\zeta_0 = (c,\ v) - \xi_0(e,\ v), \quad \xi_0 = (c,\ g).$$

XXII. Положить $\bar{w}^1 = (1,\ 0)$, $\tilde{w}^1 = (0,\ 1)$.

XXIII. Для вектора $w = (w_1,\ w_2)$ определить функцию

$$\zeta(w) = \zeta_0 + \sum_{i=1}^{n} \psi_i \times \begin{cases} \alpha_i^+, & \text{если } w_1\varphi_i + w_2\psi_i < 0, \\ \alpha_i^-, & \text{в противном случае.} \end{cases}$$

XXIV. Если $\zeta(\tilde{w}^1) > 0$, то прекратить вычисления (в этом случае исходная задача 3 не имеет решения), если $\zeta(\tilde{w}^1) < 0$, то перейти к шагу XXV.

XXV. Положить $j = 1$.

XXVI. Вычислить вектор χ^j

$$\chi^j = \frac{1}{2}(\bar{w}^j + \tilde{w}^j).$$

XXVII. Если $\zeta(\chi^j) > 0$, то положить

$$\bar{w}^{j+1} = \chi^j, \quad \tilde{w}^{j+1} = \tilde{w}^j, \quad j = j + 1,$$

и перейти к шагу XXVIII, если $\zeta(\chi^j) < 0$, то положить

$$\bar{w}^{j+1} = \bar{w}^j, \quad \tilde{w}^{j+1} = \chi^j, \quad j = j + 1$$

и перейти к шагу XXVIII.

XXVIII. Если $\| \bar{w}^j - \tilde{w}^j \| \leqslant w_0$, то перейти к шагу XXIX; иначе перейти к шагу XXVI.

XXIX. Вычислить

$$\rho = (\bar{w}_2^j + \tilde{w}_2^j)/(\bar{w}_1^j + \tilde{w}_1^j),$$

где $(\bar{w}_1^j, \bar{w}_2^j) = \bar{w}^j$; $(\tilde{w}_1^j, \tilde{w}_2^j) = \tilde{w}^j$.

XXX. Вычислить вектор

$$\tilde{g} = (1 - \rho(e, v))g + \rho v.$$

XXXI. Положить $g = \tilde{g}$ и перейти к шагу V.

Теорема 3. *Пусть основной итерационный цикл V—XXXI приводит к переходу от вектора g^k к вектору g^{k+1}. Пусть этот переход совершается в ситуации, когда приближенное решение задачи минимизации (4.42) дало точку $\tilde{x}^k$ (при переходе от шага XX к шагу XXI), удовлетворяющую условию $d \leqslant d^*$. Тогда*

$$f_0(g^{k+1}) > f_0(g^k) + \varepsilon \|\tilde{x}^k\|_G (1 - d^*),$$

где $f_0(g)$ определяется по (4.40), а норма $\| \cdot \|_G$ — по (4.41).

Алгоритм 3 за конечное число итераций находит приближенное решение $\tilde{x}$ вспомогательной задачи и отвечающее ему приближенное решение $\bar{y}$ задачи 3.

Сходимость $\| \bar{w}^j - \tilde{w}^j \|$ к нулю имеет порядок $(^1/_2)^j$, поэтому требуемая точность w_0 достигается за сравнительно небольшое число итераций по j.

4. Алгоритм «Заяц» решения прямой и двойственной задач линейного программирования

З а д а ч а 4 (прямая задача). Найти $\arg\max_{x \in X} (c, x)$ для заданного вектора $c \in R^n$ и заданного множества

$$X \triangleq \{x \mid Ax \leqslant b, \quad x \geqslant 0, \quad b \in R^m, \quad x \in R^n\}.$$

Задача 4′ (двойственная задача). Найти $\arg\min_{y \in Y}(y, b)$ для заданного вектора $b \in \boldsymbol{R}^m$ и заданного множества

$$Y \triangleq \{y \mid yA \geqslant c, \quad y \geqslant 0, \quad c \in \boldsymbol{R}^n, \quad y \in \boldsymbol{R}^m\}.$$

Предположения 4. Задачи 4 и 4′— разрешимы.

Решение пары двойственных задач 4 и 4′ сводится к решению следующей вспомогательной задачи нелинейного программирования:

найти $\arg\min_{(x, y, \xi, \eta)} g(x, y, \xi, \eta)$ при ограничениях

$$\begin{aligned} &x \geqslant 0, \quad \xi \geqslant \sigma x, \quad \xi \geqslant \tau e^n; \\ &y \geqslant 0, \quad \eta \geqslant \delta y, \quad \eta \geqslant \tau e^m, \end{aligned} \tag{4.43}$$

где

$$g(x, y, \xi, \eta) = (y, b) + (q(y), \xi) - (c, x) - (p(x), \eta);$$
$$q(y) = \max\{-yA + c, 0\};$$
$$p(x) = \max\{Ax - b, 0\};$$

$e^n \in \boldsymbol{R}^n$, $e^m \in \boldsymbol{R}^m$ — векторы с единичными компонентами; σ, τ — константы, удовлетворяющие условиям $1 < \sigma < 3$, $\tau > 0$.

Если вектор $(x^*, y^*, \xi^*, \eta^*)$ — решение вспомогательной задачи, то x^*— решение задачи 4, а y^*— решение задачи 4′.

Алгоритм 4

Начало. I. Выбрать произвольные начальные приближения $x^0 \in \boldsymbol{R}^n_+$, $y^0 \in \boldsymbol{R}^m_+$.

II. Выбрать константы σ и τ из условий

$$1 < \sigma < 3; \quad \tau > 0.$$

III. Выбрать векторы $\xi^0 = (\xi_1^0, \ldots, \xi_n^0)$, $\eta^0 = (\eta_1^0, \ldots, \eta_m^0)$, удовлетворяющие ограничениям (4.43) при $x = x^0$, $y = y^0$.

IV. Выбрать константы l и $\hat{\rho} < \frac{1}{2}$ $(l > \hat{\rho})$, определяющие шаговые множители алгоритма.

V. Выбрать константу $\varepsilon > 0$, определяющую точность вычисления оптимального значения задач 4 и 4′.

VI. Положить $x = x^0$, $y = y^0$, $\xi = \xi^0$, $\eta = \eta^0$, $\varphi_1 = -\infty$, $\psi_1 = +\infty$, $\bar{y} = y^0$, $d = 0$, $k = 1$, $\rho = 1/2$.

Основной цикл. VII. Вычислить векторы $\hat{x}(y) = (\hat{x}_1(y), \ldots, \hat{x}_n(y))$, $\hat{y}(x) = (\hat{y}_1(x), \ldots, \hat{y}_m(x))$ по формулам

$$\hat{x}_j(y) = \begin{cases} 0, & \text{если } q_j = 0, \\ \xi_j, & \text{если } q_j > 0, \end{cases} \quad j = 1, \ldots, n;$$

$$\hat{y}_\nu(x) = \begin{cases} 0, & \text{если } p_\nu = 0, \\ \eta_\nu, & \text{если } p_\nu > 0; \end{cases} \quad \nu = 1, \ldots, m,$$

где

$$p = \max\{Ax - b,\ 0\};\quad q = \max\{c - yA,\ 0\}.$$

VIII. Вычислить следующие приближения x^{k+1} и y^{k+1}

$$x^{k+1} = (1-\rho)x^k + \rho\hat{x}(y);$$

$$y^{k+1} = (1-\rho)y^k + \rho\hat{y}(x)$$

и положить $x = x^{k+1}$, $y = y^{k+1}$

IX. Если $\rho \geqslant \hat{\rho}$, то перейти к шагу X, если $\rho < \hat{\rho}$, то перейти к шагу XII.

X. Положить $k = k + 1$, $d = d + \rho$ и перейти к шагу XI.

XI. Если $d \geqslant l$, то положить $d = 0$, $\rho = \rho/2$ и перейти к шагу VII, если $d < l$, то перейти к шагу VII.

XII. Вычислить значение $\varphi(x, \eta)$, где функция $\varphi(x, \eta)$ определяется по правилу

$$\varphi(x, \eta) = (c, x) - (\eta, p),\quad p = \max\{Ax - b,\ 0\} \tag{4.44}$$

и положить $\varphi = \varphi(x, \eta)$.

XIII. Если $\varphi > \varphi_1$, то перейти к шагу XIV; иначе перейти к шагу XVII.

XIV. Положить $\varphi_1 = \varphi$, $\bar{x} = x$.

XV. Вычислить вектор

$$\xi(x) = \max\{\sigma x,\ \tau e^n\}$$

и положить $\xi = \xi(x)$.

XVI. Вычислить значение $\psi(\bar{y}, \xi)$, где функция $\psi(y, \xi)$ определяется по правилу

$$\psi(y, \xi) = (y, b) - (q, \xi),\quad q = \max\{c - yA,\ 0\} \tag{4.45}$$

и положить $\psi_1 = \psi(\bar{y}, \xi)$.

XVII. Вычислить значение $\psi(y, \xi)$, где функция $\psi(y, \xi)$ задается по (4.45), и положить $\psi = \psi(y, \xi)$.

XVIII. Если $\psi < \psi_1$, то перейти к шагу XIX; иначе перейти к шагу XXII.

XIX. Положить $\psi_1 = \psi$, $\bar{y} = y$.

XX. Вычислить вектор

$$\eta(y) = \max\{\sigma y,\ \tau e^m\}$$

и положить $\eta = \eta(y)$.

XXI. Вычислить значение $\varphi(\bar{x}, \eta)$, где функция $\varphi(x, \eta)$ задается (4.44) и положить $\varphi_1 = \varphi(\bar{x}, \eta)$.

XXII. Вычислить

$$f_k = (\varphi_1 + \psi_1)/2.$$

XXIII. Вычислить

$$\delta = (\psi_1 - \varphi_1)/(|f_k| + 1).$$

XXIV. Если $\delta \leqslant \varepsilon$, то положить $\bar{x}^k = \bar{x}$, $\bar{y}^k = \bar{y}$ и прекратить вычисления; иначе перейти к шагу X.

Теорема 4. Если выполняется предположение 4, то точки $\bar{x}^k$, $\bar{y}^k$ и значения f_k, порождаемые алгоритмом 4, удовлетворяют соотношениям

$$\lim_{\varepsilon \to 0} \bar{x}^k = x^*; \quad \lim_{\varepsilon \to 0} \bar{y}^k = y^*; \quad \lim_{\varepsilon \to 0} f_k = (c, x^*) = (b, y^*).$$

5. Итерационный метод, использующий модифицированную функцию Лагранжа для решения двойственной пары задач линейного программирования

Задача 5. Найти $\arg \max\limits_{x_j} \sum\limits_{j=1}^{n} c_j x_j$ при ограничениях

$$\sum_{j=1}^{n} \alpha_{ij} x_j \leqslant b_i, \quad i = 1, \ldots, m;$$

$$x_j \geqslant 0, \quad j = 1, \ldots, n.$$

Двойственной к задаче 5 является следующая.

Задача 5′. Найти $\arg \min\limits_{y_i} \sum\limits_{i=1}^{n} b_i y_i$ при ограничениях

$$\sum_{i=1}^{m} \alpha_{ij} y_i \geqslant c_j, \quad j = 1, \ldots, n;$$

$$y_i \geqslant 0, \quad i = 1, \ldots, m.$$

Предположение 5. Множества оптимальных решений X^* и Y^*, соответственно, задач 5 и 5′ непусты и ограничены.

В приводимом ниже алгоритме отыскание оптимальных решений X^* и Y^* задач 5 и 5′ сводится к вычислению множества седловых точек модифицированной функции Лагранжа пары двойственных задач 5 и 5′:

$$\varphi_\alpha (x, y) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} b_i y_i - \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} x_j y_i -$$

$$- \sum_{i=1}^{m} \left[\frac{\alpha}{2} ([- \varepsilon_i (x)]_+)^2 + q_\alpha (\varepsilon_i (x)) y_i \right] +$$

$$+ \sum_{j=1}^{n} \left[\frac{\alpha}{2} ([- \Delta_j (y)]_+)^2 + q_\alpha (\Delta_j (y)) x_j \right],$$

где

$$\Delta_j (y) = \sum_{i=1}^{m} \alpha_{ij} y_i - c_j, \quad j = 1, \ldots, n;$$

$$x=(x_1, \ldots, x_n), \quad y=(y_1, \ldots, y_m);$$

$$\varepsilon_i(x) = -\sum_{j=1}^{n} a_{ij}x_j + b_i, \quad i=1, \ldots, m;$$

$$[u]_+ = \begin{cases} 0, & u \leqslant 0, \\ u, & u > 0; \end{cases}$$

$$q_\alpha(t) = \begin{cases} 0, & \text{если } t \leqslant 0, \\ \alpha t^2/2, & \text{если } 0 < t \leqslant 1/\alpha, \\ t - 1/2\alpha, & \text{если } t > 1/\alpha; \end{cases}$$

$\alpha > 0$ — параметр.

При этом множество седловых точек функции $\varphi_\alpha(x, y)$ при любом $\alpha > 0$ состоит из множества $X^* \times Y^*$. Седловые точки функции $\varphi_\alpha(x, y)$ находят градиентным методом.

Отметим, что применить градиентный метод для отыскания седловых точек обычной функции Лагранжа $\varphi(x, y)$ в виде

$$\varphi(x, y) = \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} b_i y_i - \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_j y_i$$

в общем случае невозможно из-за неустойчивости множества седловых точек этой функции (определение устойчивости седловых точек приводится в § 6.9 (пункт 3)).

Алгоритм 5

Начало. I. Выбрать произвольное начальное приближение $(x^0, y^0) \in R^n \times R^m$.

II. Выбрать произвольный параметр $\alpha > 0$.

III. Положить $k = 0$.

IV. Определить функции $\Delta_j(y)$, $j = 1, \ldots, n$, и $\varepsilon_i(x)$, $i = 1, \ldots, m$, соответственно, по правилам

$$\Delta_j(y) = \sum_{i=1}^{m} a_{ij} y_i - c_j, \quad j = 1, \ldots, n, \quad (y = (y_1, \ldots, y_m));$$

$$\varepsilon_i(x) = -\sum_{j=1}^{n} a_{ij} x_j + b_i, \quad i = 1, \ldots, m, \quad (x = (x_1, \ldots, x_n)).$$

Основной цикл. V. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям теоремы 5.

VI. Вычислить вектор $\tilde{y}^k = (\tilde{y}_1^k, \ldots, \tilde{y}_m^k)$ по правилам

$$\tilde{y}_i^k = \begin{cases} y_i^k - \alpha\varepsilon_i(x^k), & \text{если } \varepsilon_i(x^k) \leqslant 0; \\ y_i^k(1 - \alpha\varepsilon_i(x^k)), & \text{если } 0 < \varepsilon_i(x^k) \leqslant 1/\alpha; \\ 0, & \text{если } \varepsilon_i(x^k) > 1/\alpha, \end{cases}$$

здесь $i = 1, \ldots, m$.

VII. Вычислить следующее приближение $x^{k+1} = (x_1^{k+1}, \ldots, x_n^{k+1})$ по правилам

$$x_j^{k+1} = \begin{cases} \max\{x_j^k - \rho_k \Delta_j(\tilde{y}^k),\ 0\}, & \text{если } \Delta_j(y^k) \leqslant 0; \\ \max\left\{x_j^k - \rho_k\left[\Delta_j(\tilde{y}^k) - \frac{\alpha}{2}(\Delta_j(y^k))^2\right],\ 0\right\}, & \text{если } 0 < \Delta_j(y^k) \leqslant 1/\alpha; \\ \max\left\{x_j^k - \rho_k\left[\Delta_j(\tilde{y}^k) - \Delta_j(y^k) + \frac{1}{2\alpha}\right],\ 0\right\}, & \text{если } \Delta_j(y^k) > 1/\alpha, \end{cases}$$

здесь $j = 1, \ldots, n$.

VIII. Вычислить вектор $\tilde{x}^k = (\tilde{x}_1^k, \ldots, \tilde{x}_n^k)$ по правилам

$$\tilde{x}_j^k = \begin{cases} x_j^k - \alpha\Delta_j(y^k), & \text{если } \Delta_j(y^k) \leqslant 0; \\ x_j^k(1 - \alpha\Delta_j(y^k)), & \text{если } 0 < \Delta_j(y^k) \leqslant 1/\alpha; \\ 0, & \text{если } \Delta_j(y^k) > 1/\alpha, \end{cases}$$

здесь $j = 1, \ldots, n$.

IX. Вычислить следующее приближение $y^{k+1} = (y_1^{k+1}, \ldots, y_m^{k+1})$ по правилам

$$y_i^{k+1} = \begin{cases} \max\{y_i^k - \rho_k \varepsilon_i(\tilde{x}^k),\ 0\}, & \text{если } \varepsilon_i(x^k) \leqslant 0; \\ \max\left\{y_i^k - \rho_k\left[\varepsilon_i(\tilde{x}^k) - \frac{\alpha}{2}(\varepsilon_i(x^k))^2\right],\ 0\right\}, & \text{если } 0 < \varepsilon_i(x^k) \leqslant 1/\alpha; \\ \max\left\{y_i^k - \rho_k\left[\varepsilon_i(\tilde{x}^k) - \varepsilon_i(x^k) + \frac{1}{2\alpha}\right],\ 0\right\}, & \text{если } \varepsilon_i(x^k) > 1/\alpha, \end{cases}$$

здесь $i = 1, \ldots, m$.

X. Положить $k = k + 1$ и перейти к шагу V.

Теорема 5. *Пусть выполняется предположение 5. Тогда существует такая константа $\rho > 0$, что если шаговые множители ρ_k в алгоритме 5 выбирать согласно условий*

$$0 < \rho_k < \rho, \quad k = 0, 1, \ldots; \quad \lim_{k\to\infty} \rho_k = 0; \quad \lim_{k\to\infty} \sum_{t=0}^{k} \rho_t = \infty,$$

то последовательности $\{x^k\}_{k=0}^{\infty}$ и $\{y^k\}_{k=0}^{\infty}$, порождаемые алгоритмом 5, при любом $\alpha > 0$ сходятся, соответственно, к X^* и Y^*.

Библиографические указания. Пункт 1 написан на основании работ [416, 535], пункт 2 — [301], пункт 3 — [363—367]. При написании пункта 4 использовалась работа [34], а пункта 5 — [82].

Итеративные методы решения задач линейного программирования изучались также в [44, 88, 85, 166, 215, 333, 359, 531, 456, 441, 5].

4.9. Методы параметрического программирования

1. Случай наличия параметра в целевой функции

Задача 1. Найти $\arg\max\limits_x \sum\limits_{j=1}^{n} (c_j^1 + \lambda c_j^2) x_j$ для заданных чисел c_j^1, c_j^2, $j = 1, \ldots, n$, при ограничениях

$$Ax = a^0;$$
$$x \geqslant 0,$$

где λ — параметр.

Предположения 1. (*i*) — множество ограничений задачи 1 непусто; (*ii*) — ранг матрицы A равен m; (*iii*) — $n > m$.

Определение 1. Базис опорного решения задачи 1 оптимален для некоторого значения λ, если оценки Δ_j всех векторов a^j, $j = 1, \ldots, n$, относительно этого базиса, вычисленные при данном λ, неотрицательны. Полная совокупность значений параметра λ, при котором базис оптимален, называется *множеством оптимальности* этого базиса.

Алгоритм, приведенный ниже, в невырожденном случае указывает (за конечное число итераций), при каких значениях $\lambda \in (-\infty, \infty)$ задача 1 неразрешима и при каких — разрешима, причем в случае разрешимости вычисляется оптимальное решение и его базис. В этом алгоритме требуется решать задачу 1 симплекс-методом при одном или нескольких значениях параметра λ, при этом исходное решение и его базис вычисляются один раз. В случае вырожденности задачи линейного программирования в алгоритме 1 теоретически возможно зацикливание (тогда надо применять специальное правило, гарантирующее от зацикливания в симплекс-методе).

Алгоритм 1

Начало. I. Выбрать произвольное число λ_0.

II. Положить $t = 0$.

Основной цикл. III. Положить $\lambda = \lambda_t$.

IV. Положить $k = 0$. Применить к решению задачи 1 симплекс-метод (если $t > 0$, то исходное опорное решение и его базис для решения задачи 1 при $\lambda = \lambda_t$ известны из предыдущих итераций).

Если задача 1 при $\lambda = \lambda_t$ разрешима, то перейти к шагу V (при этом считаются известными: оптимальный базис $a^{i_1}, \ldots, a^{i_m}$; оптимальное решение $(x_1^*, \ldots, x_n^*)$; коэффициенты разложения векторов $a^0, a^1, \ldots, a^n$ по оптимальному базису, т. е. числа z_{lj}, $j = 0, 1, \ldots, n$; $l = 1, \ldots, m$).

Если задача 1 при $\lambda = \lambda_t$ не разрешима (т. е. целевая функция задачи 1 не ограничена), то перейти к шагу XXVIII (в этом случае известны некоторое опорное решение задачи 1 и его базис $a^{i_1}, \ldots, a^{i_m}$, коэффициенты разложения z_{lj}, и, кроме того, индекс $j_t \in [1 : n]$ такой, что $\Delta_{j_t} < 0$ и при всех $l \in [1 : m]$ выполняется $z_{lj_t} \leqslant 0$).

V. Для каждого $j \in [1 : n]$ вычислить составляющие оценок Δ_j векторов a^j, $j = 1, \ldots, n$, относительно оптимального базиса $a^{i_1}, \ldots, a^{i_m}$ задачи 1 при $\lambda = \lambda_t$

$$\Delta_j = \Delta_j^1 + \lambda_t \Delta_j^2; \tag{4.46}$$

$$\Delta_j^1 = \sum_{l=1}^{m} c_{i_l}^1 z_{lj} - c_j^1, \quad j = 1, \ldots, n; \tag{4.47}$$

$$\Delta_j^2 = \sum_{l=1}^{m} c_{i_l}^2 z_{lj} - c_j^2, \quad j = 1, \ldots, n. \tag{4.48}$$

VI. Вычислить значение $\underline{\lambda}_k$ по правилу

$$\underline{\lambda}_k = \begin{cases} -\infty, & \text{если все } \Delta_j^2 \leqslant 0; \\ \max\limits_{\Delta_j^2 > 0} (-\Delta_j^1/\Delta_j^2), & \text{в противном случае.} \end{cases} \tag{4.49}$$

VII. Вычислить значение $\overline{\lambda}_k$ по правилу

$$\overline{\lambda}_k = \begin{cases} +\infty, & \text{если все } \Delta_j^2 \geqslant 0; \\ \min\limits_{\Delta_j^2 < 0} (-\Delta_j^1/\Delta_j^2), & \text{в противном случае.} \end{cases} \tag{4.50}$$

VIII. Выдать на печать: «Множество оптимальности базиса $a^{i_1}, \ldots$ $\ldots, a^{i_m}$ состоит из всех значений λ, удовлетворяющих условию $\underline{\lambda}_k \leqslant$ $\leqslant \lambda \leqslant \overline{\lambda}_k$».

Если $\overline{\lambda}_k = \infty$ и $\underline{\lambda}_k = -\infty$, то прекратить вычисления; иначе перейти к шагу IX.

IX. Если $k = 0$, то положить

$$\tilde{a}^{i_1} = a^{i_1}, \ \tilde{a}^{i_2} = a^{i_2}, \ \ldots, \ \tilde{a}^{i_m} = a^{i_m};$$

$$\tilde{\Delta}_j^1 = \Delta_j^1, \quad j = 1, \ldots, n;$$

$$\tilde{\Delta}_j^2 = \Delta_j^2, \quad j = 1, \ldots, n; \quad \tilde{\underline{\lambda}}_0 = \underline{\lambda}_0;$$

$$\tilde{z}_{lj} = z_{lj}, \quad l = 1, \ldots, m; \quad j = 0, 1, \ldots, n,$$

и перейти к шагу X; иначе перейти к шагу X.

X. Если $\overline{\lambda}_k = \infty$, то перейти к шагу XVIII; иначе вычислить индекс $s \in [1 : n]$ такой, что

$$-\frac{\Delta_s^1}{\Delta_s^2} = \overline{\lambda}_k; \ \Delta_s^2 < 0 \tag{4.51}$$

и перейти к шагу XI.

XI. Если при каждом $l \in [1 : m]$ выполняется $z_{ls} \leqslant 0$, то выдать на печать: «линейная форма задачи 1 для $\lambda > \overline{\lambda}_k$ неограничена в допустимой области» и перейти к шагу XVIII; иначе перейти к шагу XII.

XII. Вычислить индекс $r \in [1 : m]$, удовлетворяющий условию

$$z_{r0}/z_{rs} = \min_{z_{ls}>0} (z_{l0}/z_{ls}), \quad z_{rs} > 0. \qquad (4.52)$$

XIII. Перейти к новому базису, который получается заменой вектора a^{i_r} в предыдущем базисе вектором a^s (т. е. положить $i_r = s$).

XIV. Положить $\bar{z}_{lj} = z_{lj}, \quad l = 1, \ldots, m; \quad j = 0, 1, \ldots, n;$ $\bar{\Delta}_j^1 = \Delta_j^1$, $\bar{\Delta}_j^2 = \Delta_j^2$, $j = 1, \ldots, n$.

XV. Вычислить координаты всех векторов $a^0, a^1, \ldots, a^n$ в новом базисе по основным формулам:

при $l \neq r$

$$z_{lj} = \bar{z}_{lj} - (\bar{z}_{rj}/\bar{z}_{rs})\bar{z}_{ls}, \quad j = 0, 1, \ldots, n; \qquad (4.53)$$
$$l = 1, \ldots, r-1, r+1, \ldots, m,$$

при $l = r$

$$z_{lj} = \bar{z}_{rj}/\bar{z}_{rs}, \quad j = 0, 1, \ldots, n. \qquad (4.54)$$

XVI. Вычислить составляющие оценок Δ_j векторов a^j, $j = 1, \ldots, n$, относительно нового базиса $a^{i_1}, \ldots, a^{i_m}$:

$$\Delta_j^1 = \bar{\Delta}_j^1 - (\bar{z}_{rj}/\bar{z}_{rs})\bar{\Delta}_s^1, \quad j = 1, \ldots, n; \qquad (4.55)$$

$$\Delta_j^2 = \bar{\Delta}_j^2 - (\bar{z}_{rj}/\bar{z}_{rs})\bar{\Delta}_s^2, \quad j = 1, \ldots, n. \qquad (4.56)$$

XVII. Положить $k = k + 1$ и перейти к шагу VI.

XVIII. Если $\tilde{\underline{\lambda}}_0 = -\infty$, то прекратить вычисления; иначе положить $k = 0$ и перейти к шагу XIX.

XIX. Перейти к исходному оптимальному базису, исходным оценкам Δ_j^1, Δ_j^2 и исходным коэффициентам разложения z_{lj}, т. е. положить

$$a^{i_1} = \tilde{a}^{i_1}, \; a^{i_2} = \tilde{a}^{i_2}, \; \ldots, \; a^{i_m} = \tilde{a}^{i_m};$$

$$\Delta_j^1 = \tilde{\Delta}_j^1, \; \Delta_j^2 = \tilde{\Delta}_j^2, \quad j = 1, \ldots, n; \quad \underline{\lambda}_0 = \tilde{\underline{\lambda}}_0;$$

$$z_{lj} = \tilde{z}_{lj}, \quad l = 1, \ldots, m; \quad j = 0, 1, \ldots, n.$$

XX. Вычислить индекс $s \in [1 : n]$ такой, что

$$-\frac{\Delta_s^1}{\Delta_s^2} = \underline{\lambda}_k; \quad \Delta_s^2 > 0.$$

XXI. Если при каждом $l \in [1 : m]$ выполняется $z_{ls} \leqslant 0$, то выдать на печать: «Линейная форма задачи 1 для $\lambda < \underline{\lambda}_k$ неограничена в допустимой области», и прекратить вычисления; иначе перейти к шагу XXII.

XXII. Вычислить индекс $r \in [1 : m]$, удовлетворяющий условию (4.52).

XXIII. Перейти к новому базису, который получается заменой вектора a^{i_r} в предыдущем базисе вектором a^s (т. е. положить $i_r = s$).

XXIV. Положить

$$\bar{z}_{lj} = z_{lj}, \quad l = 1, \ldots, m; \quad j = 0, 1, \ldots, n;$$
$$\bar{\Delta}_j^1 = \Delta_j^1, \quad \bar{\Delta}_j^2 = \Delta_j^2, \quad j = 1, \ldots, n.$$

XXV. Вычислить величины z_{lj}, $l = 1, \ldots, m$, $j = 0, 1, \ldots, n$; Δ_j^1, $j = 1, \ldots, n$; Δ_j^2, $j = 1, \ldots, n$, соответственно, по (4.53) — (4.56).

XXVI. Вычислить значения $\underline{\lambda}_{k+1}$ и $\bar{\lambda}_{k+1}$ по формулам (4.49) и (4.50).

Выдать на печать: «Множество оптимальности базиса $a^{i_1}, \ldots, a^{i_m}$ состоит из всех значений λ, удовлетворяющих условию

$$\underline{\lambda}_{k+1} \leqslant \lambda \leqslant \bar{\lambda}_{k+1}».$$

Если $\underline{\lambda}_{k+1} = -\infty$, то прекратить вычисления; иначе перейти к шагу XXVII.

XXVII. Положить $k = k + 1$ и перейти к шагу XX.

XXVIII. Вычислить составляющие оценки Δ_{j_t}:

$$\Delta_{j_t} = \Delta_{j_t}^1 + \lambda_t \Delta_{j_t}^2;$$
$$\Delta_{j_t}^1 = \sum_{l=1}^{m} c_{i_l}^1 z_{lj_t} - c_{j_t}^1;$$
$$\Delta_{j_t}^2 = \sum_{l=1}^{m} c_{i_l}^2 z_{lj_t} - c_{j_t}^2$$

(индексы i_l, $l = 1, \ldots, m$ и j_t определены на шаге IV).

XXIX. Если $\Delta_{j_t}^2 = 0$, то выдать на печать: «Задача 1 неразрешима при всех $\lambda \in (-\infty, \infty)$» и прекратить вычисления; иначе перейти к шагу XXX.

XXX. Вычислить

$$\lambda_{t+1} = -(\Delta_{j_t}^1 / \Delta_{j_t}^2).$$

Если $\Delta_{j_t}^2 > 0$, то выдать на печать: «Задача 1 неразрешима при $\lambda < \lambda_{t+1}$» и перейти к шагу XXXI для анализа задачи при $\lambda \geqslant \lambda_{t+1}$.

Если $\Delta_{j_t}^2 < 0$, то выдать на печать: «Задача 1 неразрешима при $\lambda > \lambda_{t+1}$» и перейти к шагу XXXII для анализа задачи при $\lambda \leqslant \lambda_{t+1}$.

XXXI. Если на предыдущих итерациях был дан анализ задачи 1 для $\lambda \geqslant \lambda_{t+1}$, то прекратить вычисления; иначе положить $t = t + 1$ и перейти к шагу III.

XXXII. Если на предыдущих итерациях был дан анализ задачи 1 для $\lambda \leqslant \lambda_{t+1}$, то прекратить вычисления; иначе положить $t = t + 1$ и перейти к шагу III.

Теорема 1. *Если выполнены предположения 1, то в невырожденном случае алгоритм 1 за конечное число итераций разбивает вещественную ось λ на области, в которых задача 1 или неразрешима (вследствие неограниченности целевой функции), или имеет оптимальное решение (причем известны оптимальный базис и оптимальное опорное решение).*

2. Случай наличия параметра в правых частях ограничений

Задача 2. Найти $\arg\max\limits_{x}(c, x)$ для заданного вектора $c \in R^n$ при ограничениях

$$\sum_{j=1}^{n} \alpha_{ij}x_j = b_i^1 + \lambda b_i^2, \quad i = 1, 2, \ldots, m; \tag{4.57}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n, \tag{4.58}$$

где λ — параметр.

Предположения 2. (*i*) — целевая функция задачи 2 ограничена на допустимом множестве при всех λ; (*ii*) — ранг матрицы A равен m; (*iii*) — $n > m$.

Если при некотором значении параметра λ коэффициенты разложения вектора $b = b^1 + \lambda b^2$ (здесь $b^1 = (b_1^1, \ldots, b_m^1)$, $b^2 = (b_1^2, \ldots, b_m^2)$) по векторам базиса сопряженной задачи неотрицательны, то этот базис является оптимальным базисом задачи 2 для данного λ.

Определение 2. Полная совокупность значений параметра λ, при которых базис оптимален, называется множеством оптимальности этого базиса.

Приведенный ниже алгоритм разбивает вещественную ось λ на участки, на каждом из которых вычисляется оптимальный базис, не изменяющийся с изменением λ, а также выделяет область (если она существует), в которой ограничения задачи 2 несовместны. Алгоритм основан на применении двойственного симплекс-метода для решения задачи 2 при одном или нескольких значениях параметра λ. Алгоритм 2 применяется в невырожденном и вырожденном случаях. В вырожденном случае теоретически возможно зацикливание. Если зацикливание произойдет, то необходимо применять специальное правило, гарантирующее от зацикливания в двойственном симплекс-методе.

Алгоритм 2

Начало. I. Выбрать произвольное число λ_0.

II. Положить $t = 0$.

Основной цикл. III. Положить $\lambda = \lambda_t$.

IV. Положить $k = 0$. Применить к решению задачи 2 двойственный симплекс-метод.

Если в результате находится оптимальное решение задачи 2 и его базис (обозначить оптимальное решение через $(x_1(\lambda_t), x_2(\lambda_t), \ldots$ $\ldots, x_n(\lambda_t))$, его базис через $a^{i_1}, \ldots, a^{i_m}$, оценки через Δ_j, $j = 1, \ldots, n$, а коэффициенты разложения векторов a^j $(j = 1, \ldots, n)$ по базисным векторам через z_{lj}, $l = 1, \ldots, m$, $j = 1, \ldots, n$), то перейти к шагу V; если в результате выявится несовместность условий (4.57) и (4.58) при $\lambda = \lambda_t$, то перейти к шагу XXXI.

В этом случае, в соответствии с признаком несовместимости (см. шаг VI алгоритма 1, § 4.2) существует такое $l_t \in [1 : m]$, что $z_{l_t 0} < 0$ и при всех $j \in [1 : n]$ $z_{l_t j} \geqslant 0$, где z_{l0}, $l = 1, \ldots, m$ — коэффициенты разложения вектора $b^1 + \lambda_t b^2$ через базисные векторы $a^{i_1}, \ldots$ $\ldots, a^{i_m}$ сопряженной задачи.

V. Решить систему уравнений

$$b^1 = \sum_{l=1}^{m} a^{i_l} x^1_{i_l}(\lambda_t)$$

относительно $x^1_{i_l}(\lambda_t)$.

VI. Решить систему уравнений

$$b^1 = \sum_{l=1}^{m} a^{i_l} x^1_{i_l}(\lambda_t)$$

относительно $x^2_{i_l}(\lambda_t)$.

VII. Положить $z^1_{l0} = x^1_{i_l}(\lambda_t)$;

$$z^2_{l0} = x^2_{i_l}(\lambda_t), \quad l = 1, \ldots, m.$$

VIII. Вычислить

$$\underline{\lambda}_k = \begin{cases} -\infty, & \text{если все } z^2_{l0} \leqslant 0, \quad (l = 1, \ldots, m), \\ \max\limits_{z^2_{l0} > 0} (-z^1_{l0}/z^2_{l0}), & \text{если существует } z^2_{l0} > 0. \end{cases} \tag{4.59}$$

IX. Вычислить значение $\overline{\lambda}_k$ по правилу

$$\overline{\lambda}_k = \begin{cases} +\infty, & \text{если все } z^2_{l0} \geqslant 0, \\ \min\limits_{z^2_{l0} < 0} (-z^1_{l0}/z^2_{l0}), & \text{если существует } z^2_{l0} < 0, \end{cases} \tag{4.60}$$

здесь $l = 1, \ldots, m$.

X. Выдать на печать: «Множество оптимальности базиса a^{i_1}, $a^{i_2}, \ldots, a^{i_m}$ состоит из всех значений λ, удовлетворяющих условию $\underline{\lambda}_k \leqslant \lambda \leqslant \overline{\lambda}_k$» и перейти к шагу XI.

XI. Если $\underline{\lambda}_k = -\infty$ и $\overline{\lambda}_k = +\infty$, то прекратить вычисления; иначе перейти к шагу XII.

XII. Если $k = 0$, то положить

$$\tilde{a}^{i_1} = a^{i_1}, \ \ldots, \ \tilde{a}^{i_m} = a^{i_m};$$

$$\tilde{\Delta}_j = \Delta_j, \quad j = 1, \ \ldots, \ n; \quad \tilde{\lambda}_0 = \underline{\lambda}_0;$$

$$\tilde{z}_{lj} = z_{lj}, \quad l = 1, \ \ldots, \ m; \quad j = 1, \ \ldots, \ n;$$

$$\tilde{z}^1_{l0} = z^1_{l0}, \quad \tilde{z}^2_{l0} = z^2_{l0}, \quad l = 1, \ \ldots, \ m,$$

и перейти к шагу XIII; иначе перейти к шагу XIII.

XIII. Если $\bar{\lambda}_k = +\infty$, то перейти к шагу XXI; иначе вычислить индекс $r \in [1 : m]$ такой, что

$$-z^1_{r0}/z^2_{r0} = \bar{\lambda}_k; \quad z^2_{r0} < 0.$$

XIV. Вычислить индекс $s \in [1 : n]$ такой, что

$$-\Delta_s/z_{rs} = \min_{z_{rj}<0} (-\Delta_j/z_{rj}). \tag{4.61}$$

XV. Положить

$$\bar{z}_{lj} = z_{lj}, \quad l = 1, \ \ldots, \ m; \quad j = 0, \ 1, \ \ldots, \ n;$$

$$\bar{\Delta}_j = \Delta_j, \quad j = 1, \ \ldots, \ n;$$

$$\bar{z}^1_{l0} = z^1_{l0}, \quad \bar{z}^2_{l0} = z^2_{l0}, \quad l = 1, \ \ldots, \ m.$$

XVI. Перейти к новому базису $a^{i_1}, \ldots, a^{i_m}$, который получается заменой вектора a^{i_r} в предыдущем базисе вектором a^s (т. е. положить $i_r = s$).

XVII. Вычислить координаты всех векторов $a^1, \ldots, a^n$ в новом базисе по основным формулам:

при $l \neq r$

$$z_{lj} = \bar{z}_{lj} - (\bar{z}_{rj}/\bar{z}_{rs})\, \bar{z}_{ls}, \quad j = 1, \ \ldots, \ n; \tag{4.62}$$
$$l = 1, \ \ldots, \ r-1, \ r+1, \ \ldots, \ m,$$

при $l = r$

$$z_{lj} = \bar{z}_{rj}/\bar{z}_{rs}, \quad j = 1, \ \ldots, \ n. \tag{4.63}$$

XVIII. Вычислить оценки

$$\Delta_j = \bar{\Delta}_j - (\bar{z}_{rj}/\bar{z}_{rs})\, \bar{\Delta}_s, \quad j = 1, \ \ldots, \ n. \tag{4.64}$$

XIX. Вычислить величины z^1_{l0} и z^2_{l0}, $l = 1, \ldots, m$, по основным формулам:

$$z^1_{l0} = \bar{z}^1_{l0} - (\bar{z}^1_{r0}/\bar{z}_{rs})\, \bar{z}_{ls}, \quad l = 1, \ \ldots, \ r-1, \ r+1, \ \ldots, \ m; \quad z^1_{r0} = \bar{z}^1_{r0}/\bar{z}_{rs}; \tag{4.65}$$

$$z^2_{l0} = \bar{z}^2_{l0} - (\bar{z}^2_{r0}/\bar{z}_{rs})\, \bar{z}_{ls}, \quad l = 1, \ \ldots, \ r-1, \ r+1, \ \ldots, \ m; \quad z^2_{r0} = \bar{z}^2_{r0}/\bar{z}_{rs}. \tag{4.66}$$

XX. Положить $k = k + 1$ и перейти к шагу VIII.

XXI. Если $\tilde{\underline{\lambda}}_0 = -\infty$, то прекратить вычисления; иначе положить $k = 0$ и перейти к шагу XXII.

XXII. Перейти к исходному оптимальному базису, исходным оценкам, исходным коэффициентам разложения, т. е. положить

$$a^{i_1} = \tilde{a}^{i_1},\ a^{i_2} = \tilde{a}^{i_2},\ \ldots,\ a^{i_m} = \tilde{a}^{i_m};$$

$$\Delta_j = \tilde{\Delta}_j,\quad j = 1, \ldots, n;\quad \underline{\lambda}_0 = \tilde{\underline{\lambda}}_0;$$

$$z_{lj} = \tilde{z}_{lj},\quad j = 1, \ldots, n;\quad l = 1, \ldots, m;$$

$$z^1_{l0} = \tilde{z}^1_{l0},\quad z^2_{l0} = \tilde{z}^2_{l0},\quad l = 1, \ldots, m.$$

XXIII. Вычислить индекс $r \in [1 : m]$ такой, что

$$-z^1_{r0}/z^2_{r0} = \underline{\lambda}_k \quad \text{и} \quad z^2_{r0} > 0.$$

XXIV. Вычислить индекс $s \in [1 : n]$, удовлетворяющий условию (4.61).

XXV. Положить

$$\bar{z}_{lj} = z_{lj},\quad l = 1, \ldots, m,\quad j = 1, \ldots, n;$$

$$\bar{\Delta}_j = \Delta_j,\quad j = 1, \ldots, n;$$

$$\bar{z}^1_{l0} = z^1_{l0},\ \bar{z}^2_{l0} = z^2_{l0},\quad l = 1, \ldots, m.$$

XXVI. Перейти к новому базису, который получается заменой вектора a^{i_r} предыдущего базиса вектором a^s (т. е.положить $i_r = s$).

XXVII. Вычислить величины

$$z_{lj},\ l = 1, \ldots, m;\quad j = 1, \ldots, n;$$

$$\Delta_j,\ j = 1, \ldots, n;\quad z^1_{l0},\quad l = 1, \ldots, m;\quad z^2_{l0},\ l = 1, \ldots, m,$$

соответственно по (4.62) — (4.66).

XXVIII. Вычислить значения $\underline{\lambda}_{k+1}$ и $\bar{\lambda}_{k+1}$ по (4.59) и (4.60) соответственно. Выдать на печать: «Множество оптимальности базиса $a^{i_1}, a^{i_2}, \ldots, a^{i_m}$ состоит из всех значений λ, удовлетворяющих условию $\underline{\lambda}_{k+1} \leqslant \lambda \leqslant \bar{\lambda}_{k+1}$».

XXIX. Если $\underline{\lambda}_{k+1} = -\infty$, то прекратить вычисления; иначе перейти к шагу XXX.

XXX. Положить $k = k + 1$ и перейти к шагу XXIII.

XXXI. Вычислить величины z^1_{l0}, z^2_{l0}, $l = 1, \ldots, m$, по тем правилам, что и на шагах V, VI, VII.

XXXII. Если $z^2_{l_t 0} = 0$, то выдать на печать: «Ограничения задачи 2 несовместимые при всех $\lambda \in (-\infty, \infty)$» и прекратить вычисления; иначе перейти к шагу XXXIII.

XXXIII. Вычислить

$$\lambda_{t+1} = -(z^1_{i_t 0}/z^2_{i_t 0}).$$

Если $z^2_{i_t 0} > 0$, то выдать на печать: «Ограничения задачи 2 несовместимые при всех $\lambda < \lambda_{t+1}$» и перейти к шагу XXXIV для анализа задачи при $\lambda \geqslant \lambda_{t+1}$, если $z^2_{i_t 0} < 0$, то выдать на печать: «Ограничения задачи 2 несовместимые при всех $\lambda > \lambda_{t+1}$» и перейти к шагу XXXV для анализа задачи при $\lambda \leqslant \lambda_{t+1}$.

XXXIV. Если на предыдущих итерациях был дан анализ задачи 2 для $\lambda \geqslant \lambda_{t+1}$, то прекратить вычисления; иначе положить $t = t + 1$ и перейти к шагу III.

XXXV. Если на предыдущих итерациях был дан анализ задачи 2 для $\lambda \leqslant \lambda_{t+1}$, то прекратить вычисления; иначе положить $t = t + 1$ и перейти к шагу III.

Теорема 2. *Если выполнены предположения 2, то в невырожденном случае алгоритм 2 за конечное число итераций разбивает вещественную ось λ на области, в которых или задача 2 имеет оптимальный базис и оптимальное решение, или ограничения задачи 2 несовместимы.*

Библиографические указания. При написании параграфа использовались работы [88, 226] и др. Дополнительные сведения о методах параметрического программирования можно найти в работах [310, 311, 190, 477, 478, 554, 475].

4.10. Опорные методы

Задача 0. Найти $\arg\max_{x \in X} (c, x)$ для заданного вектора $c \in \in \boldsymbol{R}^n$ и заданного множества

$$X \triangleq \{x \mid Ax = a^0, \quad x \geqslant 0, \quad x \in \boldsymbol{R}^n\}.$$

Предположения 0. (*i*) — ранг матрицы A равен m; (*ii*) — $m \leqslant \leqslant n$; (*iii*) — задача 0 имеет допустимые решения.

Опорные методы занимают промежуточное положение между конечными и итерационными методами решения задач линейного программирования. Вместо понятий «опорное решение» и «базис опорного решения» используются их обобщения — «допустимое решение» и «опора». В приводимых алгоритмах процесс решения задачи можно вести или до получения оптимального решения, или остановить на ε-оптимальном решении, отклонение которого от оптимального по целевой функции не превосходит заданной величины $\varepsilon > 0$. Начиная с некоторого номера итерации опорного метода совпадают с итерациями симплекс-метода. Достоинством опорных методов является то, что они позволяют эффективно использовать хорошие начальные приближения, интуицию и опыт специалистов. Другими достоинствами является более простое отыскание начального опорного плана по сравнению с отысканием опорного решения и его базиса.

Определение 1. Совокупность векторов $a^{i_1}, \ldots, a^{i_m}$ называется опорой задачи 0, если уравнение $\sum_{s=1}^{m} a^{i_s} x_{i_s} = 0$ (a^i, $i = 1, \ldots, n$,— столбцы матрицы A) имеет только нулевое решение $x_{i_s} = 0$, $s = 1, \ldots, m$, но для любого вектора a^i, $i = 1, \ldots, n$, $i \neq i_s$, $s = 1, \ldots, m$, уравнение $\sum_{s=1}^{m} a^{i_s} x_{i_s} + a^i x_i = 0$ имеет не нулевое решение.

Введем обозначения: $\mathcal{J}_{оп} \triangleq \{i_1, \ldots, i_m\}$ — множество опорных индексов; $\mathcal{J}_н \triangleq \mathcal{J} \setminus \mathcal{J}_{оп}$— множество неопорных индексов (здесь $\mathcal{J} = \{1, \ldots, n\}$).

Определение 2. Пару $\{x, A_{оп}\}$, состоящую из допустимого решения x и опоры $A_{оп}$, называют *опорным планом.*

Через $A_{оп}$ обозначена матрица, состоящая из столбцов $a^{i_1}, \ldots, a^{i_m}$. Отличие опорного плана от опорного решения и его базиса состоит в том, что неопорные компоненты x_i, $i \in \mathcal{J}_н$ опорного плана не обязательно равны нулю.

Определение 3. Опорный план $\{x, A_{оп}\}$ называют *невырожденным*, если все его опорные компоненты $x_{i_1}, \ldots, x_{i_m}$ положительные.

1. Прямой опорный метод

Алгоритм 1

Начало. I. Найти невырожденный опорный план $\{x, A_{оп}\}$ задачи 0

$$x = (x_1, \ldots, x_n), \quad A_{оп} = \{a^{i_1}, \ldots, a^{i_m}\}.$$

II. Выбрать число $\varepsilon > 0$ — точность вычисления максимума целевой функции задачи 0.

III. Вычислить матрицу $A_{оп}^{-1}$, обратную исходной опорной матрице.

IV. Найти коэффициенты разложения z_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$ по опоре $A_{оп}$ векторов a^j, $j = 1, \ldots, n$ (т. е. найти числа z_{ij}, такие, что

$$a^j = \sum_{s=1}^{m} a^{i_s} z_{sj}),$$

по формулам

$$\begin{gathered}
(z_{1j}, \ldots, z_{mj})^T = A_{оп}^{-1} (\alpha_{1j}, \ldots, \alpha_{mj})^T, \\
j = 1, \ldots, n, \quad j \neq i_s, \quad s = 1, \ldots, m; \\
z_{li_s} = 0 \quad \text{при} \quad s \neq l; \\
z_{li_s} = 1 \quad \text{при} \quad s = l, \quad s = 1, \ldots, m; \quad l = 1, \ldots, m.
\end{gathered} \tag{4.67}$$

Основной цикл. V. Положить $\mathcal{J}_{оп} = \{i_1, \ldots, i_m\}$, $\mathcal{J}_н = \mathcal{J} \setminus \mathcal{J}_{оп}$.

VI. Разбить множество индексов $\mathcal{J}_н$ на два непересекающихся подмножества $\mathcal{J}_{н+}$ и $\mathcal{J}_{н0}$ следующим образом: индекс $j \in \mathcal{J}_н$ входит в $\mathcal{J}_{н+}$ тогда и только тогда, когда $x_j > 0$; индекс $j \in \mathcal{J}_н$ входит в $\mathcal{J}_{н0}$ тогда и только тогда, когда $x_j = 0$.

VII. Вычислить оценки Δ_j, $j = 1, ..., n$, по формулам

$$\Delta_j = \sum_{k=1}^{m} c_{i_k} z_{kj} - c_j, \quad j = 1, \ldots, n; \quad j \neq i_k, \quad k = 1, \ldots, m;$$

$$\Delta_j = 0, \quad j = i_k, \quad k = 1, \ldots, m.$$

VIII. Если выполняются условия

$$\Delta_j = 0 \quad \text{при} \quad j \in \mathcal{J}_{н+};$$

$$\Delta_j \geqslant 0 \quad \text{при} \quad j \in \mathcal{J}_{н0},$$

то положить $x^* = x$ и прекратить вычисления (в этом случае допустимое решение x является оптимальным); иначе перейти к шагу IX.

IX. Если выполняются условия

$$\Delta_j \geqslant 0 \quad \text{при} \quad j = 1, \ldots, n;$$

$$\sum_{j=1}^{n} \Delta_j x_j \leqslant \varepsilon,$$

то положить $x^\varepsilon = x$ и прекратить вычисления (в этом случае допустимое решение x является ε-оптимальным или субоптимальным решением); иначе перейти к шагу X. (Допустимое решение x^ε называется ε-оптимальным (субоптимальным) решением, если $(c, x^*) - (c, x^\varepsilon) \leqslant \varepsilon$).

X. Если существует индекс $\hat{j} \in [1 : n]$ такой, что $\Delta_{\hat{j}} < 0$ и $z_{i\hat{j}} \leqslant 0$ при всех $i \in [1 : m]$, то прекратить вычисления (в этом случае целевая функция задачи 0 не ограничена сверху на множестве X); иначе перейти к шагу XI.

XI. Найти оценку Δ_{j_0}, на которой достигается максимум в выражении

$$\max \{\max_{x_j > 0} |\Delta_j|, \quad \max_{x_j = 0} (-\Delta_j)\}.$$

XII. Если $\Delta_{j_0} < 0$, то перейти к шагу XIII; если $\Delta_{j_0} > 0$, то перейти к шагу XVIII.

XIII. Вычислить параметр

$$\theta_0 = \min_{\substack{z_{kj_0} > 0 \\ 1 \leqslant k \leqslant m}} (x_{i_k} / z_{kj_0})$$

и найти индекс $r \in [1 : m]$ такой, что

$$x_{i_r} / z_{rj_0} = \theta_0, \quad z_{rj_0} > 0.$$

XIV. Положить

$$\bar{x}_j = x_j, \quad j = 1, \ldots, n;$$

$$\bar{z}_{ij} = z_{ij}, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

XV. Перейти к новому опорному плану $\{x, A_{оп}\}$ по следующим правилам:

$$x_{i_k} = \bar{x}_{i_k} - \theta_0 z_{kj_0} \quad \text{при} \quad k = 1, \ldots, m;$$

$$x_j = \bar{x}_j \quad \text{при} \quad j \neq i_k, \quad k = 1, \ldots, m; \quad j \neq j_0;$$

$$x_{j_0} = \bar{x}_{j_0} + \theta_0;$$

$i_r = j_0$ (остальные опорные индексы не меняют своих значений).

XVI. Найти коэффициенты разложения z_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, векторов a^j, $j = 1, \ldots, n$, по новой опоре $A_{оп}$:

$$z_{ij} = \bar{z}_{ij} - (\bar{z}_{rj}\bar{z}_{ij_0}/\bar{z}_{rj_0}), \quad i \neq r, \quad i = 1, \ldots, m; \quad j = 1, \ldots, n,$$

$$z_{rj} = \bar{z}_{rj}/\bar{z}_{rj_0}.$$

XVII. Перейти к шагу V.

XVIII. Вычислить параметр

$$\theta_0' = \min_{\substack{z_{kj_0} < 0 \\ 1 \leqslant k \leqslant m}} (-x_{i_k}/z_{kj_0}).$$

XIX. Найти индекс $r \in [1 : m]$ такой, что

$$\theta_0' = -x_{i_r}/z_{rj_0}, \quad z_{rj_0} < 0.$$

XX. Вычислить параметр

$$\theta_0 = \min\{\theta_0', x_{j_0}\}.$$

XXI. Положить

$$\bar{x}_j = x_j, \quad j = 1, \ldots, n;$$

$$\bar{z}_{ij} = z_{ij}, \quad i = 1, \ldots, m; \quad j = 1, \ldots, n.$$

XXII. Если $\theta_0 = \theta_0' \neq x_{j_0}$, то перейти к шагу XXIII, если $\theta_0 = x_{j_0}$ или $\theta_0' = x_{j_0}$, то перейти к шагу XXVI.

XXIII. Перейти к новому опорному плану $\{x, A_{оп}\}$ по следующим правилам:

$$x_{i_k} = \bar{x}_{i_k} + \theta_0 z_{kj_0} \quad \text{при} \quad k = 1, \ldots, m;$$

$$x_j = \bar{x}_j \quad \text{при} \quad j \neq i_k, \quad k = 1, \ldots, m; \quad j \neq j_0;$$

$$x_{j_0} = \bar{x}_{j_0} - \theta_0;$$

$i_r = j_0$ (остальные опорные индексы $i_1, \ldots, i_{r-1}, i_{r+1}, \ldots, i_m$ не меняют своих значений).

XXIV. Вычислить коэффициенты разложения z_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, векторов a^j, $j = 1, \ldots, n$ по новой опоре $A_{оп}$ по формулам (4.67).

XXV. Перейти к шагу V.

XXVI. Перейти к новому опорному плану $\{x, A_{оп}\}$ по следующим правилам:

$$x_{i_k} = \bar{x}_{i_k} + \bar{x}_{j_0} z_{kj_0} \quad \text{при} \quad k = 1, \ldots, m;$$
$$x_{j_0} = 0;$$
$$x_j = \bar{x}_j \quad \text{при} \quad j \neq i_k, \quad k = 1, \ldots, m; \quad j \neq j_0,$$

опорные индексы не меняют своих значений (а значит, не меняются числа z_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$).

XXVII. Перейти к шагу V.

Теорема 1. *Если выполнены предположения 0 и задача 0 невырождена, то процесс решения задачи 0 алгоритмом 1 через конечное число итераций закончится либо на шаге VIII (находится оптимальное решение x^* задачи 0), либо на шаге IX (находится ε-оптимальное решение задачи 0), либо на шаге X (устанавливается неограниченность целевой функции задачи 0 на допустимом множестве X).*

2. Метод обратной матрицы

В прямом опорном методе на каждой итерации запоминается $n \times m$ чисел z_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, что требует значительной оперативной памяти ЭВМ. В методе обратной матрицы вместо чисел z_{ij} запоминается матрица $A_{оп}^{-1}$ размера $m \times m$, обратная к опорной матрице, что экономит оперативную память ЭВМ.

Алгоритм 2

Начало. I. Найти невырожденный опорный план $\{x, A_{оп}\}$ задачи 0:

$$x = (x_1, \ldots, x_n), \quad A_{оп} = (a^{i_1}, \ldots, a^{i_m}).$$

II. Выбрать число $\varepsilon > 0$ — точность вычисления максимума целевой функции задачи 0.

III. Вычислить матрицу $A_{оп}^{-1} \triangleq (u_{lk})_{l=1,\ldots,m}^{k=1,\ldots,m}$, обратную к исходной опорной.

Основной цикл. IV. Положить $c_{баз} = (c_{i_1}, \ldots, c_{i_m})$,

$$A_{оп}^{-1} = (u^1, \ldots, u^m) = \begin{pmatrix} v^1 \\ \vdots \\ v^m \end{pmatrix}$$

(здесь $u^1, \ldots, u^m$ — столбцы; $v^1, \ldots, v^m$ — строки).

V. Вычислить вектор $b = (b_1, \ldots, b_m)$:

$$b = c_{баз} A_{оп}^{-1}.$$

VI. Вычислить оценки Δ_j, $j = 1, \ldots, n$, по правилам

$$\Delta_j = ba^j - c_j, \quad j \neq i_k, \quad k = 1, \ldots, m;$$
$$\Delta_j = 0, \quad j = i_k, \quad k = 1, \ldots, m.$$

VII. Если выполняются условия

$$\Delta_j = 0 \text{ при всех } x_j > 0;$$
$$\Delta_j \geqslant 0 \text{ при всех } x_j = 0,$$

то положить $x^* = x$ и прекратить вычисления (в этом случае допустимое решение x является оптимальным); иначе перейти к шагу VIII.

VIII. Если выполняются условия

$$\Delta_j \geqslant 0 \quad \text{при} \quad j = 1, \ldots, n;$$
$$\sum_{j=1}^{n} \Delta_j x_j \leqslant \varepsilon,$$

то положить $x^\varepsilon = x$ и прекратить вычисления (в этом случае допустимое решение x является ε-оптимальным); иначе перейти к шагу IX.

IX. Положить $t = 1$.

X. Если $\Delta_t < 0$, то вычислить числа

$$z_{it} = v^i a^t, \quad i = 1, \ldots, m,$$

и перейти к шагу XI; иначе перейти к шагу XII.

XI. Если при всех $i = 1, \ldots, m$ выполняется неравенство $z_{it} \leqslant \leqslant 0$, то прекратить вычисления (в этом случае целевая функция задачи 0 неограничена сверху на допустимом множестве X); иначе перейти к шагу XII.

XII. Если $t < n$, то положить $t = t + 1$ и перейти к шагу X; иначе перейти к шагу XIII.

XIII. Найти индекс $j_0 \in [1 : n]$, имеющий оценку Δ_{j_0}, при которой обеспечивается максимум выражению

$$\max \{\max_{x_j > 0} |\Delta_j|, \max_{x_j = 0} (-\Delta_j)\}.$$

XIV. Если $\Delta_{j_0} < 0$, то перейти к шагу XV, если $\Delta_{j_0} > 0$, то перейти к шагу XXI.

XV. Вычислить величины

$$z_{ij_0} = v^i a^{j_0}, \quad i = 1, \ldots, m.$$

XVI. Вычислить параметр

$$\theta_0 = \min_{z_{kj_0} > 0} (x_{i_k} / z_{kj_0})$$

и найти индекс $r \in [1 : m]$ такой, что

$$\theta_0 = x_{i_r} / z_{rj_0}, \ z_{rj_0} > 0.$$

XVII. Положить

$$\bar{x}_j = x_j, \quad j = 1, \ldots, n;$$
$$\bar{u}_{lk} = u_{lk}, \quad l = 1, \ldots, m, \quad k = 1, \ldots, m,$$

где $u^k = (u_{1k}, u_{2k}, \ldots, u_{mk})^T$.

XVIII. Перейти к новому опорному плану $\{x, A_{\text{оп}}\}$ по следующим правилам:

$$x_{i_k} = \bar{x}_{i_k} - \theta_0 z_{kj_0} \quad \text{при} \quad k = 1, \ldots, m;$$

$$x_j = \bar{x}_j \quad \text{при} \quad j \neq i_k, \quad k = 1, \ldots, m, \quad j \neq j_0;$$

$$x_{j_0} = \bar{x}_{j_0} + \theta_0;$$

$i_r = j_0$ (остальные опорные индексы не меняют своих значений).

XIX. Вычислить матрицу $A_{\text{оп}}^{-1} \triangleq (u_{lk})_{l=1,\ldots,m}^{k=1,\ldots,m}$, обратную новой опорной матрице, по формулам

$$u_{lk} = \bar{u}_{lk} - \bar{u}_{rk} z_{lj_0}/z_{rj_0}, \quad l = 1, \ldots, r-1, r+1, \ldots, m;$$
$$k = 1, \ldots, m; \qquad (4.68)$$
$$u_{rk} = \bar{u}_{rk}/z_{rj_0}, \quad k = 1, \ldots, m.$$

XX. Перейти к шагу IV.

XXI. Вычислить величины

$$z_{ij_0} = v^i a^{j_0}, \quad i = 1, \ldots, m.$$

XXII. Вычислить параметр

$$\theta_0' = \min_{\substack{z_{kj_0} < 0 \\ 1 \leqslant k \leqslant m}} (-x_{i_k}/z_{kj_0})$$

и найти индекс $r \in [1:m]$ такой, что

$$\theta_0' = -x_{i_r}/z_{rj_0}, \quad z_{rj_0} < 0.$$

XXIII. Вычислить параметр

$$\theta_0 = \min\{\theta_0', x_{j_0}\}.$$

XXIV. Положить

$$\bar{x}_j = x_j, \quad j = 1, \ldots, n;$$

$$\bar{u}_{lk} = u_{lk}, \quad l = 1, \ldots, m; \quad k = 1, \ldots, m.$$

XXV. Если $\theta_0 = \theta_0' \neq x_{j_0}$, то перейти к шагу XXVI, если $\theta_0 = x_{j_0}$ или $\theta_0' = x_{j_0}$, то перейти к шагу XXIX.

XXVI. Перейти к новому опорному плану $\{x, A_{\text{оп}}\}$ по следующим правилам:

$$x_{i_k} = \bar{x}_{i_k} + \theta_0 z_{kj_0} \quad \text{при} \quad k = 1, \ldots, m;$$

$$x_j = \bar{x}_j \quad \text{при} \quad j \neq i_k, \quad k = 1, \ldots, m, \quad j \neq j_0;$$

$$x_{j_0} = \bar{x}_{j_0} - \theta_0;$$

$i_r = j_0$ (остальные опорные индексы $i_1, \ldots, i_{r-1}, i_{r+1}, \ldots, i_m$ не меняют своих значений).

XXVII. Вычислить матрицу $A_{\text{оп}}^{-1} \triangleq (u_{lk})_{l=1,\ldots,m}^{k=1,\ldots,m}$, обратную новой опорной матрице, по (4.68).

XXVIII. Перейти к шагу IV.

XXIX. Перейти к новому опорному плану $\{x, A_{оп}\}$ по следующим правилам:

$$x_{i_k} = \bar{x}_{i_k} + \bar{x}_{j_0} z_{kj_0} \quad \text{при} \quad k = 1, \ldots, m;$$

$$x_j = \bar{x}_j \quad \text{при} \quad j \neq i_k, \quad k = 1, \ldots, m, \quad j \neq j_0;$$

$x_{j_0} = 0$ (опорные индексы $i_1, \ldots, i_m$, не меняют своих значений, а значит, не меняется матрица $A_{оп}^{-1}$).

XXX. Перейти к шагу IV.

Теорема 2. *Если выполнены предположения 0 и задача 0 невырождена, то решение задачи 0 алгоритмом 2 через конечное число итераций закончится либо на шаге VII (находится оптимальное решение x^* задачи 0), либо на шаге VIII (находится ε-оптимальное решение задачи 0), либо на шаге XI (устанавливается неограниченность целевой функции задачи 0 на допустимом множестве X).*

Замечание 1. Если в алгоритме 2 (на шагах XIX и XXVII) матрицу $A_{оп}^{-1}$, обратную новой опорной, вычислять по формуле $A_{оп}^{-1} = D_r \bar{A}_{оп}^{-1}$, где $\bar{A}_{оп}^{-1} \triangleq (\bar{u}_{lk})_{l=1,\ldots,m}^{k=1,\ldots,m}$ — матрица, обратная «старой» опорной матрице, и

$$D_r = \begin{pmatrix} 1 & \ldots & 0 & -z_{1j_0}/z_{rj_0} & 0 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \ldots & 1 & -z_{r-1,j_0}/z_{rj_0} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1/z_{rj_0} & 0 & \ldots & 0 \\ 0 & \ldots & 0 & -z_{r+1,j_0}/z_{rj_0} & 0 & \ldots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \ldots & 0 & -z_{mj_0}/z_{rj_0} & 0 & \ldots & 1 \end{pmatrix},$$

то такой метод решения задачи называется *мультипликативным*. Для записи матрицы D_r достаточно знать $m + 1$ чисел, т. е. число r и числа, которые стоят в r-м столбце. Если обозначить матрицу D_r в k-й итерации через D_{r_k}, а за исходную опору в алгоритме 2 выбрать единичную матрицу I, то обратная к опорной в k-й итерации будет определяться по формуле

$$A_{оп}^{-1} = D_{r_k} D_{r_{k-1}} \ldots D_{r_1} I.$$

Отсюда следует, что для определения матрицы $A_{оп}^{-1}$ в k-й итерации требуется знать $(m + 1) \times k$ чисел, что на начальных стадиях мультипликативного алгоритма приводит к экономии оперативной памяти ЭВМ.

Библиографические указания. Параграф написан на основании работы [59]. В указанной работе приведены также различные модификации прямых опорных методов, двойственные опорные методы, безопорные методы, комбинированные методы.

Глава 5

МЕТОДЫ РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО И СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

5.1. Методы проекции градиента

1. Общий алгоритм

Задача 1. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества $X \subset R^n$.

Предположение 1. (*i*) — функция f_0 выпукла вниз и непрерывно дифференцируема на множестве X; (*ii*) — X — выпуклое замкнутое множество; (*iii*) —

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma \|x - y\|, \quad 0 < \gamma < \infty, \quad \forall x, y \in X.$$

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить $\nabla f_0(x^k)$.

IV. Выбрать множитель β_k, удовлетворяющий одному из заранее выбранных условий

$$0 < \beta' \leqslant \beta_k \leqslant \beta'' < +\infty; \tag{5.1}$$

$$0 < \beta' \leqslant \beta_k \leqslant \beta'' < 2/\gamma. \tag{5.2}$$

V. Вычислить точку $y^k \in R^n$ по формуле

$$y^k = x^k - \beta_k \nabla f_0(x^k).$$

VI. Вычислить точку y_X^k — проекцию точки y^k на выпуклое замкнутое множество X.

VII. Если β_k выбирается в соответствии с неравенствами (5.2), то выбрать множитель ρ_k, удовлетворяющий неравенствам

$$0 < \hat{\rho} < \rho_k \leqslant 1 \tag{5.3}$$

и перейти к шагу VIII, если β_k выбирается согласно (5.1), то вычислить множитель ρ_k по одной из формул

$$f_0(x^k - \rho_k(x^k - y_X^k)) = \min_{0\leqslant\rho\leqslant 1} f_0(x^k - \rho(x^k - y_X^k)), \tag{5.4}$$

или

$$\rho_k = \min\left\{1,\ \nu_k \frac{(\nabla f_0(x^k),\ x^k - y_X^k)}{\|x^k - y_X^k\|^2}\right\}, \tag{5.5}$$

где $0 < \varepsilon_1 < \nu_k \leqslant \dfrac{2-\varepsilon_2}{\gamma}$; $\varepsilon_2 > 0$,

и перейти к шагу VIII.

VIII. Положить $x^{k+1} = x^k - \rho_k(x^k - y_X^k)$, $k = k + 1$ и перейти к шагу III.

Предполагается при этом, что ρ_k на каждой итерации вычисляется по одной и той же формуле.

Теорема 1. *Пусть выполняются предположения 1 и (iv) — множество*

$$X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0),\ x \in X\}$$

ограничено;
(v) —

$$\|\nabla f_0(x)\| \leqslant \tau < +\infty \ \text{для всех}\ x \in X_0.$$

Тогда алгоритм 1 порождает последовательность $\{x^k\}_{k=0}^{\infty}$ *такую, что*

$$f_0(x^k) - \min_{x \in X} f_0(x) \leqslant \alpha/k, \quad k = 0,\ 1,\ \ldots,\ \alpha > 0.$$

Причем: а) если в алгоритме 1 β_k *удовлетворяет неравенствам (5.1), а* ρ_k *— равенству (5.4), то справедлива оценка*

$$f_0(x^k) - f_0(x^{k+1}) \geqslant \left(\frac{1}{\beta''}\bar\rho - \frac{\gamma}{2}\bar\rho^2\right)\|x^k - y_X^k\|^2, \quad k = 0,\ 1,\ \ldots,$$

где $0 < \bar\rho < \min\left\{1 - \frac{2}{\gamma\beta''}\right\}$;

б) если в алгоритме 1 β_k *удовлетворяет неравенствам (5.1), а* ρ_k *вычисляется по (5.5), то справедлива оценка: при* $\rho_k = 1$

$$f_0(x^k) - f_0(x^{k+1}) \geqslant \frac{\varepsilon_2}{2\beta''}\|x^k - y_X^k\|^2, \quad k = 0,\ 1,\ \ldots;$$

при $\rho_k = \nu_k(\nabla f_0(x^k),\ x^k - y_X^k)/\|x^k - y_X^k\|^2$

$$f_0(x^k) - f_0(x^{k+1}) \geqslant \frac{\varepsilon_1\varepsilon_2}{2(\beta'')^2}\|x^k - y_X^k\|^2, \quad k = 0,\ 1,\ \ldots;$$

в) если в алгоритме 1 β_k *удовлетворяет неравенствам (5.2), а* ρ_k *— (5.3), то справедлива оценка*

$$f_0(x^k) - f_0(x^{k+1}) \geqslant \hat\rho\left(\frac{1}{\beta''} - \frac{\gamma}{2}\right)\|x^k - y_X^k\|^2.$$

2. Метод проекции градиента для минимизации функций при линейных ограничениях

Задача 2. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного множества

$$X = \{x \mid (a^j,\ x) - b_j \leqslant 0,\quad j = 1,\ \ldots,\ m,\quad x \in \boldsymbol{R}^n\},$$

где $a^j \in \boldsymbol{R}^n$; $b_j \in \boldsymbol{R}^1$; $j = 1,\ \ldots,\ m$.

Предположение 2. Функция f_0 выпукла и непрерывно дифференцируема.

Обозначения и определения. 1. Произвольное подмножество множества $\{1,\ \ldots,\ m\}$, содержащее m' элементов, обозначим через $\mathcal{J}$.

2. Матрицу размера $n \times m'$, столбцами которой являются векторы a^j, $j \in \mathcal{I}$, расположенные в порядке возрастания j, обозначим через $A_{\mathcal{I}}$.

3. Матрицу проектирования $\boldsymbol{R}^n$ на подпространство $L_{\mathcal{I}}$, порожденное векторами a^j, $j \in \mathcal{I}$, обозначим через $P_{\mathcal{I}}$.

4. Если векторы a^j, $j \in \mathcal{I}$, определяющие подпространство $L_{\mathcal{I}}$, линейно независимы, то матрица, проектирующая $\boldsymbol{R}^n$ на подпространство $L_{\mathcal{I}}$, вычисляется по формуле

$$P_{\mathcal{I}} = A_{\mathcal{I}} (A_{\mathcal{I}}^T A_{\mathcal{I}})^{-1} A_{\mathcal{I}}^T. \tag{5.6}$$

Если векторы a^j, $j \in \mathcal{I}$ линейно зависимы, то необходимо найти подмножество $\mathcal{I}'$ такое, что векторы a^j, $j \in \mathcal{I}'$ будут линейно независимы и порождают $L_{\mathcal{I}}$, а проектирующая матрица $P_{\mathcal{I}} = P_{\mathcal{I}'}$ будет определяться (5.6) при $\mathcal{I} = \mathcal{I}'$.

5. Матрицу проектирования $\boldsymbol{R}^n$ на ортогональное дополнение к $L_{\mathcal{I}}$ обозначим через

$$P_{\mathcal{I}}^{\perp} = I - P_{\mathcal{I}},$$

где I — единичная $n \times n$-матрица.

6. Обозначим через $\mathcal{I}_\varepsilon(x)$ ε-активное множество индексов, определяемое формулой

$$\mathcal{I}_\varepsilon(x) = \{j \mid (a^j, x) - b_j + \varepsilon \geqslant 0, \quad j = 1, \ldots, m, \quad x \in X\},$$

где $\varepsilon \geqslant 0$.

Утверждение 2. Точка $\hat{x} \in X$ оптимальна в задаче 2 тогда и только тогда, когда

$$\nabla f_0(\hat{x}) = A_{\mathcal{I}_0(\hat{x})} y^0(\hat{x}), \text{ т. е. } P^{\perp}_{\mathcal{I}_0(\hat{x})} \nabla f_0(\hat{x}) = 0;$$

$$y^0(\hat{x}) \leqslant 0,$$

где $y^0(\hat{x})$ вычисляется по следующей формуле (при $\varepsilon = 0$ и $x = \hat{x}$):

$$y^\varepsilon(x) = (A^T_{\mathcal{I}_\varepsilon(x)} A_{\mathcal{I}_\varepsilon(x)})^{-1} A^T_{\mathcal{I}_\varepsilon(x)} \nabla f_0(x). \tag{5.7}$$

Заметим, что матрица $P_{\mathcal{I}_\varepsilon(x)}$, проектирующая $\boldsymbol{R}^n$ на подпространство $L_{\mathcal{I}_\varepsilon(x)}$, натянутое на векторы a^j, $j \in \mathcal{I}_\varepsilon(x)$, вычисляется по формуле

$$P_{\mathcal{I}_\varepsilon(x)} = A_{\mathcal{I}_\varepsilon(x)} (A^T_{\mathcal{I}_\varepsilon(x)} A_{\mathcal{I}_\varepsilon(x)})^{-1} A^T_{\mathcal{I}_\varepsilon(x)},$$

если выполнено предположение 2'.

Предположение 2'. Существует такое число $\varepsilon' > 0$, что для любой точки $x \in X$ и любого числа $\varepsilon \in [0, \varepsilon']$ векторы a^j, $j \in \mathcal{I}_\varepsilon(x)$ линейно независимы.

Алгоритм 2

Начало. I. Вычислить $x^0 \in X$; выбрать $\beta \in (0, 1)$ (обычно полагают $\beta = 1/2$), $\bar{\varepsilon} \in (0, \varepsilon')$ и $\varepsilon'' \in (0, \bar{\varepsilon})$; положить $k = 0$.

Основной цикл. II. Положить $x = x^k$.

III. Положить $\varepsilon_0 = \bar{\varepsilon}$, $j = 0$.

IV. Вычислить множество индексов $\mathcal{J}_{\varepsilon_j}(x)$ и положить $\mathcal{J} = \mathcal{J}_{\varepsilon_j}(x)$.

V. Вычислить матрицу $P_{\mathcal{J}}^{\perp}$, проектирующую R^n на ортогональное дополнение к $L_{\mathcal{J}}$,

$$P_{\mathcal{J}}^{\perp} = I - A_{\mathcal{J}} (A_{\mathcal{J}}^T A_{\mathcal{J}})^{-1} A_{\mathcal{J}}^T. \tag{5.8}$$

VI. Вычислить вектор

$$h^{\varepsilon_j}(x) = P_{\mathcal{J}}^{\perp} \nabla f_0(x). \tag{5.9}$$

VII. Если $\| h^{\varepsilon_j}(x) \|^2 > \varepsilon_j$, то положить $h(x) = -h^{\varepsilon_j}(x)$ и перейти к шагу XVII; иначе перейти к шагу VIII.

VIII. Если $\varepsilon_j \leqslant \varepsilon''$, то вычислить множество индексов $\mathcal{J}_0(x)$ и перейти к шагу IX; иначе перейти к шагу XII.

IX. Вычислить вектор

$$h^0(x) = P_{\mathcal{J}_0}^{\perp} \nabla f_0(x), \tag{5.10}$$

где $P_{\mathcal{J}_0}^{\perp}$ — матрица, определяемая по (5.8) при $\mathcal{J} = \mathcal{J}_0$.

X. Вычислить вектор $y^0(x)$ по (5.7) при $\varepsilon = 0$.

XI. Если $\| h^0(x) \|^2 = 0$ и $y^0(x) \leqslant 0$, то положить $x^* = x$ и прекратить вычисления; иначе перейти к шагу XII.

XII. Вычислить вектор $y^{\varepsilon_j}(x)$ по формуле (5.7) при $\varepsilon = \varepsilon_j$ и положить $y = y^{\varepsilon_j}(x)$.

XIII. Если $y \leqslant 0$, то положить $\varepsilon_{j+1} = \beta\varepsilon_j$, $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу XIV.

XIV. Предполагая, что $\mathcal{J} = \{i_1, i_2, \ldots, i_{m'}\}$ и $i_1 < i_2 < \ldots < i_{m'}$, положить $\mu_{i_\alpha}(x) = y_\alpha$, $\alpha = 1, \ldots, m'$.

XV. Найти наименьшее число i среди чисел $l \in \mathcal{J}$, для которых вектор

$$\bar{h}^{\varepsilon_j}(x) \triangleq P_{\mathcal{J}-i}^{\perp} \nabla f_0(x) \tag{5.11}$$

удовлетворяет условию

$$\| \bar{h}^{\varepsilon_j}(x) \| = \max \{ \| P_{\mathcal{J}-l}^{\perp} \nabla f_0(x) \| \mid l \in \mathcal{J}, \quad \mu_l(x) > 0 \}, \tag{5.12}$$

(здесь и далее $\mathcal{J} - i \triangleq \{ j \mid j \in \mathcal{J}, \quad j \neq i \}$) и положить $h(x) = -\bar{h}^{\varepsilon_j}(x)$.

XVI. Если $\| h(x) \|^2 \leqslant \varepsilon_j$, то положить $\varepsilon_{j+1} = \beta\varepsilon_j$, $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу XVII.

XVII. Вычислить наименьшее из чисел $\rho(x) > 0$, удовлетворяющих условию

$$f_0(x+\rho(x)h(x)) = \min\{f_0(x+\rho h(x)) \mid \rho \geqslant 0,\ x+\rho h(x) \in X\}. \quad (5.13)$$

XVIII. Вычислить следующее приближение

$$x^{k+1} = x + \rho(x)h(x).$$

XIX. Положить $k = k+1$ и перейти к шагу II.

Теорема 2. *Если выполнены предположения 2, 2' и множество*

$$X'(x^0) \triangleq \{x \mid f_0(x) \leqslant f_0(x^0),\quad (a^j, x) - b_j \leqslant 0,\quad j = 1, \ldots, m\}$$

компактно, то последовательность x^k, $k = 0, 1, \ldots$, построенная алгоритмом 2, либо конечна и ее последний элемент оптимален для задачи 2, либо бесконечна и каждая ее предельная точка оптимальна для задачи 2.

Замечание 2. Если шаг XIX алгоритма 2 заменить шагом XIX' или XIX'', описанными ниже, то заключения теоремы 2 остаются в силе при условии, что величина $\rho(x)$ на каждой итерации однозначно определена. (Такая замена является рациональной с вычислительной точки зрения).

XIX'. Положить $\bar{\varepsilon} = \varepsilon_j$, $k = k+1$ и перейти к шагу II.

XIX''. Если k/i' — целое число (здесь целое число $i' \geqslant 1$ введено для управления скоростью уменьшения величины $\bar{\varepsilon}$), то положить $\bar{\varepsilon} = \varepsilon_j$, $k = k+1$ и перейти к шагу II; иначе положить $k = k+1$ и перейти к шагу II.

Замечание 2'. Чтобы получить реализуемую версию алгоритма 2, можно шаг XVII алгоритма заменить на XVII'. Вычислить наименьшее из целых чисел $i \geqslant 0$, для которых

$$f_0(x+\bar{\beta}^i\rho h(x)) - f_0(x) - \bar{\beta}^i\rho\alpha(\nabla f_0(x),\ h(x)) \leqslant 0;$$

$$f_l(x+\bar{\beta}^i\rho h(x)) \leqslant 0,\quad l = 1, 2, \ldots, m;$$

$$f_l(z) \equiv (a^l, z) - b_l,$$

и положить $\rho(x) = \bar{\beta}^i\rho$, где $\rho > 0$; $\bar{\beta} \in (0, 1)$; $\alpha \in (0, 1)$.

***Алгоритм 2'* (ускоренная версия алгоритма 2)**

Начало. I. Вычислить $x^0 \in X$; выбрать $\beta \in (0, 1)$, $\bar{\varepsilon} \in (0, \varepsilon')$ и $\varepsilon'' \in (0, \bar{\varepsilon})$; положить $k = 0$.

Основной цикл. II. Положить $x = x^k$.

III. Положить $\varepsilon_0 = \bar{\varepsilon}$, $j = 0$.

IV. Вычислить множество индексов $\mathcal{J}_{\varepsilon_j}(x)$ и положить $\mathcal{J} = \mathcal{J}_{\varepsilon_j}(x)$.

V. Вычислить проектирующую матрицу $P_{\mathcal{J}}^{\perp}$ по (5.8).

VI. Вычислить вектор $h^{\varepsilon_j}(x)$ по (5.9).

VII. Если $\| h^{\varepsilon_j}(x)\|^2 > \varepsilon_j$, то перейти к шагу VIII; иначе перейти к шагу XII.

VIII. Вычислить вектор $y^{\varepsilon_j}(x)$ в соответствии с (5.7) и положить $y = y^{\varepsilon_j}(x)$.

IX. Если $y \leqslant 0$, то положить $h(x) = -h^{\varepsilon_j}(x)$ и перейти к шагу XIX; иначе перейти к шагу X.

X. Предполагая, что $\mathcal{J} = \{i_1, i_2, \ldots, i_{m'}\}$ и $i_1 < i_2 < \ldots < i_{m'}$, положить $\mu_{i_\alpha}(x) = y_\alpha$, $\alpha = 1, \ldots, m'$.

XI. Вычислить $\bar{h}^{\varepsilon_j}(x)$ согласно (5.11) и (5.12), положить $h(x) = -\bar{h}^{\varepsilon_j}(x)$ и перейти к шагу XIX.

XII. Если $\varepsilon_j < \varepsilon''$, то вычислить векторы $h^0(x)$, $y^0(x)$ согласно (5.10) и (5.7) и перейти к шагу XIII; иначе перейти к шагу XIV.

XIII. Если $\|h^0(x)\|^2 = 0$ и $y^0(x) \leqslant 0$, то положить $x^* = x$ и прекратить вычисления; иначе перейти к шагу XIV.

XIV. Вычислить вектор $y^{\varepsilon_j}(x)$ согласно (5.7) и положить $y = y^{\varepsilon_j}(x)$.

XV. Если $y \leqslant 0$, то положить $\varepsilon_{j+1} = \beta\varepsilon_j$, $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу XVI.

XVI. Предполагая, что $\mathcal{J} = \{i_1, i_2, \ldots, i_{m'}\}$ и $i_1 < i_2 < \ldots < i_{m'}$, положить $\mu_{i_\alpha}(x) = y_\alpha$, $\alpha = 1, 2, \ldots, m'$.

XVII. Вычислить вектор $\bar{h}^{\varepsilon_j}(x)$ согласно (5.11) и (5.12), положить $h(x) = -\bar{h}^{\varepsilon_j}(x)$.

XVIII. Если $\|h(x)\|^2 \leqslant \varepsilon_j$, то положить $\varepsilon_{j+1} = \beta\varepsilon_j$, $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу XIX.

XIX. Вычислить наименьшее из чисел $\rho(x) \in (0, \infty)$, удовлетворяющих (5.13).

XX. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho(x)\, h(x).$$

XXI. Положить $k = k + 1$ и перейти к шагу II.

Для последовательности x^k, $k = 0, 1, \ldots$, порожденной алгоритмом 2′, справедлива теорема 2.

3. Гибридный метод проекции градиента для минимизации функций при нелинейных ограничениях

Задача 3. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}.$$

Предположения 3. (*i*) — функции f_j, $j = 0, 1, \ldots, m$, непрерывно дифференцируемы и выпуклы; (*ii*) — существует такое число $\varepsilon' > 0$, что для любой точки $x \in X$ и любого числа $\varepsilon \in [0, \varepsilon']$ векторы $\nabla f_j(x)$, $j \in \mathcal{J}_\varepsilon(x)$, линейно независимы.

Гибридный метод проекции градиента объединяет идеи метода проекции градиента (алгоритм 2) и метода возможных направлений (алгоритм 1 см. § 5.7). Приводимый ниже метод получен в [536]

добавлением к методу (описанному в [500]) ε-процедуры, которая сделала этот метод заведомо сходящимся (версия метода, приведенная в [500], таким свойством не обладает).

Алгоритм 3

Начало. I. Вычислить $x^0 \in X$; выбрать $\beta \in (0, 1)$ (обычно полагают $\beta = 1/2$), $\bar{\varepsilon} \in (0, \varepsilon')$ и $\varepsilon'' \in (0, \bar{\varepsilon})$; положить $k = 0$.

Основной цикл. II. Положить $x = x^k$.

III. Положить $\varepsilon_0 = \bar{\varepsilon}$, $j = 0$.

IV. Вычислить ε_j-активное множество индексов $\mathcal{J}_{\varepsilon_j}(x)$ по формуле

$$\mathcal{J}_{\varepsilon_j}(x) = \{i \mid f_i(x) + \varepsilon_j \geqslant 0, \quad i = 1, \ldots, m, \quad x \in X\} \tag{5.14}$$

и положить $\mathcal{J} = \mathcal{J}_{\varepsilon_j}(x)$.

V. Найти матрицу $A_{\mathcal{J}}$, столбцами которой являются векторы $\nabla f_j(x)$, $j \in \mathcal{J}$, линейно упорядоченные по j.

VI. Вычислить проектирующую матрицу

$$P_{\mathcal{J}}^{\perp} = I - A_{\mathcal{J}}(A_{\mathcal{J}}^T A_{\mathcal{J}})^{-1} A_{\mathcal{J}}^T.$$

VII. Вычислить вектор

$$h^{\varepsilon_j}(x) = P_{\mathcal{J}}^{\perp} \nabla f_0(x).$$

VIII. Если $\|h^{\varepsilon_j}(x)\|^2 > \varepsilon_j$, то положить $h(x) = -h^{\varepsilon_j}(x)$ и перейти к шагу XX; иначе перейти к шагу IX.

IX. Если $\varepsilon_j \leqslant \varepsilon''$, то вычислить множество индексов $\mathcal{J}_0(x)$ по (5.14) (при $\varepsilon_j = 0$) и перейти к шагу X; иначе перейти к шагу XV.

X. Найти матрицу $A_{\mathcal{J}_0}$ (как на шаге V при $\mathcal{J} = \mathcal{J}_0$).

XI. Вычислить проектирующую матрицу

$$P_{\mathcal{J}_0}^{\perp} = I - A_{\mathcal{J}_0}(A_{\mathcal{J}_0}^T A_{\mathcal{J}_0})^{-1} A_{\mathcal{J}_0}^T.$$

XII. Вычислить вектор

$$h^0(x) = P_{\mathcal{J}_0}^{\perp} \nabla f_0(x).$$

XIII. Вычислить вектор

$$y^0(x) = (A_{\mathcal{J}_0(x)}^T A_{\mathcal{J}_0(x)})^{-1} A_{\mathcal{J}_0(x)}^T \nabla f_0(x).$$

XIV. Если $\|h^0(x)\|^2 = 0$ и $y^0(x) \leqslant 0$, то положить $x^* = x$ и прекратить вычисления; иначе перейти к шагу XV.

XV. Вычислить вектор $y^{\varepsilon_j}(x)$ согласно (5.7) при $\varepsilon = \varepsilon_j$ и положить $y = y^{\varepsilon_j}(x)$.

XVI. Если $y \leqslant 0$, то положить $\varepsilon_{j+1} = \beta\varepsilon_j$, $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу XVII.

XVII. Предполагая, что $\mathcal{J} = \{i_1, i_2, \ldots, i_{m'}\}$ и $i_1 < i_2 < \ldots < i_{m'}$, положить $\mu_{i_\alpha}(x) = y_\alpha$, $\alpha = 1, 2, \ldots, m'$.

XVIII. Найти наименьшее из чисел $i \in \mathcal{I}$, для которых вектор

$$\bar{h}^{\varepsilon_j}(x) \triangleq P^{\perp}_{\mathcal{I}-i}\nabla f_0(x)$$

удовлетворяет соотношению

$$\|\bar{h}^{\varepsilon_j}(x)\| = \max\{\|P^{\perp}_{\mathcal{I}-l}\nabla f_0(x)\| \mid l \in \mathcal{I}, \quad \mu_l(x) > 0\},$$

и положить $h(x) = -\bar{h}^{\varepsilon_j}(x)$.

XIX. Если $\|h(x)\|^2 < \varepsilon_j$, то положить $\varepsilon_{j+1} = \beta\varepsilon_j$, $j = j + 1$ и перейти к шагу IV; иначе перейти к шагу XX.

Примечание. В общем случае луч $\{x' = x + \rho h(x) \mid \rho \geqslant 0\}$ пересекает множество X только в точке x, т. е. вектор $h(x)$ не определяет возможное направление. Ниже строится вектор $q(x)$, определяющий возможное направление.

XX. Если $h(x) = -h^{\varepsilon_j}(x)$, то положить $\mathcal{K} = \mathcal{I}$ и перейти к шагу XXI; иначе положить $\mathcal{K} = \mathcal{I} - i$ и перейти к шагу XXI.

XXI. Вычислить вектор

$$q(x) = \lambda(x)h(x) + A_{\mathcal{K}}(A^T_{\mathcal{K}}A_{\mathcal{K}})^{-1}g,$$

где $g = -\varepsilon_j(1, 1, \ldots, 1) \in R^n$ и $\lambda(x) \geqslant 1$ — наименьшее из чисел на луче $[1, \infty)$, для которых

$$(\nabla f_l(x),\ q(x)) \leqslant -\varepsilon_j,$$

где $l = 0$ при $\mathcal{K} = \mathcal{I}$ и $l = i$ при $\mathcal{K} = \mathcal{I} - i$.

XXII. Вычислить наименьшее из чисел $\rho(x) > 0$, удовлетворяющих условию

$$f_0(x + \rho(x)q(x)) = \min\{f_0(x + \rho q(x)) \mid \rho \geqslant 0,\ x + \rho q(x) \in X\}.$$

XXIII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho(x)q(x).$$

XXIV. Положить $k = k + 1$ и перейти к шагу II.

Теорема 3. *Пусть выполнены предположения 3 и пусть задача 3 такова, что алгоритм 3 корректен для всех*

$$x \in \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in X\}.$$

Тогда последовательность x^k, $k = 0, 1, \ldots$, порожденная алгоритмом 3, либо конечна и ее конечный элемент оптимален для задачи 3, либо бесконечна и каждая ее предельная точка оптимальна для задачи 3.

Замечание 3. Алгоритм 3 можно привести к реализуемой форме тем же способом, что и алгоритм 2 (см. замечание 2′). Можно также получить ускоренную версию алгоритма 3, используя для этого идею построения алгоритма 2′.

Задача 4. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества X.

Предположения 4. (*i*) — X — замкнутое выпуклое множество; (*ii*) — f_0— дифференцируемая на X функция.

В алгоритме 4 предполагается, что имеется возможность вычислять на каждой итерации приближенное значение h^k градиента функции f_0 в точке x^k.

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить направление движения h^k (приближенно совпадающее с градиентом функции f_0 в точке $x = x^k$).

IV. Найти шаговый множитель ρ_k.

V. Вычислить следующее приближение

$$x^{k+1} = \pi_X(x^k - \rho_k h^k),$$

где π_X — оператор проектирования на множество X.

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 4. *Пусть выполняются предположения 4 и* (*iii*) — *градиент функции* f_0 *на множестве*

$$X_0 = \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in X\}$$

удовлетворяет условию Липшица с константой γ *(здесь* x^0 *— начальное приближение в алгоритме 4);* (*iv*) *— вектор* h^k*, вычисляемый на шаге III алгоритма 4, удовлетворяет условию*

$$\|h^k - \nabla f_0(x^k)\| \leqslant \beta \|x^{k+1} - x^k\|, \quad k = 0, 1, \ldots .$$

Тогда для последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 4, справедливы утверждения:*

1) *если* $0 \leqslant \rho_k \leqslant 2/(\gamma + 2\beta)$, $k = 0, 1, \ldots$, *то* $\{f_0(x^k)\}_{k=0}^{\infty}$ *монотонно убывает;*

2) *если* X_0 *ограничено,* f_0 *выпуклая функция,*

$$0 < \varepsilon_1 \leqslant \rho_k \leqslant 2/(\gamma + 2\beta) - \varepsilon_2, \quad k = 0, 1, \ldots, \quad (\varepsilon_2 > 0), \text{ то}$$

$$\lim_{k\to\infty} f_0(x^k) = f_0^* \triangleq \inf_{x\in X} f_0(x), \quad f_0(x^k) - f_0^* = O\left(\frac{1}{k}\right);$$

3) *если функция* f_0 *дважды дифференцируема и*

$$\gamma_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x)\, y, y) \leqslant \gamma \|y\|^2, \ \gamma_1 > 0;$$

$$\|\nabla^2_{xx} f_0(x) - \nabla^2_{xx} f_0(y)\| \leqslant \gamma_2 \|x - y\|;$$

$$\delta = \max\{|1 - \rho\gamma_1|, \ |1 - \rho\gamma|\}, \quad q = \frac{\delta + \beta\rho}{1 - \beta\rho} < 1,$$

то при $\rho_k = \rho$, $k = 0, 1, \ldots$, *последовательность* $\{x^k\}_{k=0}^{\infty}$ *сходится к точке минимума* x^*, *причем*

$$\|x^k - x^*\| = O(q^k).$$

Библиографические указания. Пункт 1 написан на основании работы [192], пункт 2 и 3 — на основании работ [285, 510], пункт 4 — на основании работы [295]. Различные модификации метода проекции градиента и различные способы построения проектирующих матриц описаны также в [551, 500, 320, 536, 25].

5.2. Общий метод штрафных функций

Задача 1. Найти $\arg\max_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X = \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m; \quad x \in Y \subset R^n\}.$$

Предположения 1. (*i*) — Y — компактное множество в R^n; (*ii*) — функции f_j, $j = 0, 1, \ldots, m$ — непрерывны на множестве Y.

Определение 1. Семейство функций $\psi(\cdot, \alpha)$, определенных на множестве Y, зависящих от параметра α и обладающих свойствами: (*i*) — $\psi(x, \alpha) \geqslant 0$ при всех $x \in Y$; (*ii*) — $\psi(x, \alpha) \to 0$ (монотонно убывая) при $\alpha \to \infty$ на множестве X^0, всюду плотном в X; (*iii*) — $\lim_{\alpha \to \infty} \psi(x, \alpha) = \infty$ равномерно по $x \in (Y \setminus V_\delta(X))$ для любого $\delta > 0$ (где $V_\delta(X)$ обозначает δ-окрестность множества X) называют *штрафными функциями* (или *функциями штрафа*), а параметр α — *штрафным коэффициентом* (или *коэффициентом штрафа*).

Если $\psi(\cdot, \alpha)$ — штрафные функции, то справедливо

$$\max_{x \in X} f_0(x) = \lim_{\alpha \to \infty} \sup_{x \in Y} (f_0(x) - \psi(x, \alpha)),$$

и, таким образом, решение задачи 1 сводится к вспомогательным задачам максимизации по $x \in Y$ функции $f_0(x) - \psi(x, \alpha)$ при достаточно больших α. Решение последней в силу «простой» структуры множества Y является проще исходной задачи. В общем случае решение задачи 1 находится как предельное решение вспомогательных задач при $\alpha \to \infty$. В некоторых случаях решение исходной задачи можно получить при конечном значении α.

Ниже приведены примеры штрафных функций:

$$\psi(x, \alpha) = \alpha \sum_{i=1}^{m} |\min\{0, f_i(x)\}|^\tau, \quad \tau > 0; \tag{5.14}$$

$$\psi(x, \alpha) = \alpha |\min\{0, \min_{i \in [1:m]} f_i(x)\}|^\tau, \quad \tau > 0; \tag{5.15}$$

$$\psi(x, \alpha) = \begin{cases} 0, & \text{если } \min_{i \in [1:m]} f_i(x) \geqslant 0, \\ \alpha e^{1/\min_i f_i(x)} & \text{в противном случае;} \end{cases} \tag{5.16}$$

$$\psi(x, \alpha) = \left[1 + \sum_{i=1}^{m} |\min\{0, f_i(x)\}|^{\tau}\right]^{\alpha} - 1, \quad \tau > 0; \qquad (5.17)$$

$$\psi(x, \alpha) = \begin{cases} \frac{1}{\alpha} \sum_{i=1}^{m} f_i^{-1}(x) \text{ при } \min_{i \in [1:m]} f_i(x) > 0, \\ +\infty, \text{ в противном случае;} \end{cases} \qquad (5.18)$$

$$\psi(x, \alpha) = \sum_{i=1}^{m} \exp(-\alpha f_i(x)). \qquad (5.19)$$

Штрафные функции (5.14) — (5.17) называют *внешними*, а (5.18) — (5.19) — *внутренними штрафными функциями*.

Для функции $\psi(x, \alpha)$ вида

$$\psi(x, \alpha) = \begin{cases} -\frac{1}{\alpha} \sum_{i=1}^{m} \ln f_i(x), & x \in X^0; \\ \infty, & x \notin X^0 \end{cases}$$

условие (*i*) в определении 1 штрафных функций в общем случае не выполняется, но теорема 1 остается для нее справедливой.

Алгоритм 1

I. Выбрать семейство штрафных функций $\psi(\cdot, \alpha)$.

II. Задать последовательность $\{\alpha_k\}_{k=0}^{\infty}$ штрафных коэффициентов такую, что $\lim_{k \to \infty} 1/\alpha_k = 0$.

III. Задать последовательность $\{\varepsilon_k\}_{k=0}^{\infty}$ такую, что

$$\lim_{k \to \infty} \varepsilon_k = 0, \quad \varepsilon_k > 0, \quad k = 0, 1, \ldots .$$

IV. Вычислить при каждом k ($k = 0, 1, \ldots$) точку x^k, удовлетворяющую условию

$$f_0(x^k) - \psi(x^k, \alpha_k) \geqslant \sup_{x \in Y} (f_0(x) - \psi(x, \alpha_k)) - \varepsilon_k.$$

Здесь также требуется решать задачу условной максимизации. Однако на практике под множеством Y обычно подразумевается некоторое «простое» множество (например, параллелепипед в пространстве R^n) и решение этой задачи с точки зрения наличия ограничений не представляет особых трудностей.

Теорема 1. *Если выполняются предположения 1, то любая предельная точка последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, принадлежит допустимому множеству* X *и реализует* $\max_{x \in X} f_0(x)$, *т. е. является решением задачи 1.*

В следующей теореме приводятся оценки скорости сходимости алгоритма 1, на основании которых выбираются штрафные коэффициенты, обеспечивающие заданную точность решения по функционалу.

Теорема 1'. *Пусть выполняются предположения 1 и* (*iii*) — *функция* $f_0(x)$ *удовлетворяет условию Липшица на компакте* Y *с*

константой γ; (*iv*) — *существуют постоянные* δ, β, $\sigma > 0$ *такие, что при всех* $x \in (V_\delta(X) \setminus X) \cap Y$ *справедливо неравенство*

$$\min_{i\in[1:m]} f_i(x) \leqslant -\beta[\rho_Y(x, X)]^\sigma,$$

где ρ_Y — *метрика в* Y; $V_\delta(X)$ — δ-*окрестность множества* X; (*v*) — *штрафная функция* $\psi(x, \alpha)$ *имеет вид*

$$\psi_\tau(x, \alpha) = \alpha \sum_{i=1}^{m} |\min\{0, f_i(x)\}|^\tau, \quad \tau > 0.$$

Тогда для достаточно больших α *при* $\tau\sigma > 1$ *справедливы неравенства*

$$0 \leqslant \Delta(\alpha, \tau) \leqslant \mu(\gamma^{\tau\sigma}/\alpha)^{1/(\tau\sigma-1)};$$

$$\rho_Y(x^*(\alpha, \tau), X) \leqslant \mu(\gamma/\alpha)^{1/(\tau\sigma-1)},$$

где

$$\Delta(\alpha, \tau) = \max_{x\in Y}(f_0(x) - \psi_\tau(x, \alpha)) - \max_{x\in X} f_0(x);$$

μ — *величина, не зависящая от* α, γ;

$$x^*(\alpha, \tau) = \arg\max_{x\in Y}(f_0(x) - \psi_\tau(x, \alpha)).$$

Если $\tau\sigma \leqslant 1$, *то существует значение штрафного коэффициента*

$$\bar{\alpha} = \bar{\alpha}(\gamma, \beta, \tau) = O(\gamma/\beta^\tau)$$

такое, что $x^*(\alpha, \tau) \in X$ *и* $\Delta(\alpha, \tau) = 0$ *при всех* $\alpha \geqslant \bar{\alpha}(\gamma, \beta, \tau)$.

Замечание 1. Условие (*iv*) теоремы 1′ с произвольным $\delta > 0$, $\sigma = 1$ и некоторым $\beta > 0$ выполняется в каждом из следующих двух случаев:

1) функции $f_i(x)$, $i = 1, \ldots, m$ — вогнуты на компакте Y и выполнено условие Слейтера, т. е. существует точка $\bar{x} \in Y$ такая, что $f_i(\bar{x}) > 0$, $i = 1, \ldots, m$;

2) Y — выпуклое множество из $\boldsymbol{R}^n$, а $f_i(x)$, $i = 1, \ldots, m$,— линейные функции.

Замечание 1′. Условие (*iv*) теоремы 1′ с произвольным $\sigma > 0$, $\delta > 0$ и некоторым (достаточно малым) β выполняется, если Y — конечное множество.

Замечание 1″. Если система неравенств

$$f_i(x) \geqslant 0, \quad i = 1, \ldots, m,$$

имеет единственное решение $x^* \in \boldsymbol{R}^n$, причем

$$f_1(x^*) = \cdots = f_{n+1}(x^*) = 0;$$

$$f_{n+2}(x^*) > 0, \ldots, f_m(x^*) > 0$$

и любые n векторов системы $\{\nabla f_1(x^*), \ldots, \nabla f_{n+1}(x^*)\}$ линейно независимы, то

$$\min_{1\leqslant i\leqslant m} f_i(x) \leqslant -\beta\|x - x^*\|.$$

Замечание 1‴. Из теоремы 1′ следует, что при $\tau\sigma \leqslant 1$ возможно точное решение задачи математического программирования методом штрафных функций при конечном значении коэффициента штрафа α. Обеспечить выполнение условия $\tau\sigma \leqslant 1$ можно выбором параметра τ. Однако следует учитывать, что при $\tau \leqslant 1$ функция $\psi_\tau(x, \alpha)$ вообще не является дифференцируемой по x, даже если функции $f_i(x)$, $i = 1, \ldots, m$ дифференцируемы по x, что может усложнить решение задачи $\max\limits_{x \in Y} (f_0(x) - \psi_\tau(x, \alpha))$.

Библиографические указания. При написании параграфа использовались работы [370, 372]. Дополнительные сведения об общем методе штрафов, оценках скорости сходимости, рекомендациях по практическому использованию метода штрафов можно найти в работах [141, 146, 289, 89, 468, 571, 452, 514, 430].

В работе [565] доказывается, что при линейных внешних штрафных функциях из существования седловой точки для функции Лагранжа вытекает сходимость решений вспомогательных задач безусловной оптимизации к решению исходной задачи при конечном значении коэффициента штрафа.

В работе X. Майна и М. Фукушима [525] для построения штрафных функций используется понятие инфимальной сходимости последовательности функций и доказывается, что некоторые известные методы внутренних и внешних штрафных функций являются частными случаями их теории.

5.3. Методы внешних штрафных функций

1. Задачи с ограничениями в виде неравенств

Задача 1. Найти $\arg\min\limits_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и для заданного множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j = 1, 2, \ldots, m, \quad x \in \boldsymbol{R}^n\}. \tag{5.20}$$

Предположения 1. (*i*) — функция f_0 непрерывно дифференцируема; (*ii*) — функции f_j, $j = 1, 2, \ldots, m$ непрерывны; (*iii*) — существует такая точка $x' \in X$, что множество

$$X' = \{x \mid f_0(x) \leqslant f_0(x')\}$$

компактно.

Идея метода внешних штрафных функций заключается в сведении решения задачи с ограничениями к решению последовательности задач на безусловный минимум. При этом необходимо строить последовательность внешних штрафных функций для множества X, которая определяется следующим образом.

Определение 1. Последовательность непрерывных функций $p_k(x)$, $k = 0, 1, 2, \ldots$, называется последовательностью внешних штрафных функций для множества X, если:

$$\begin{array}{lll} p_k(x) = 0 & \text{для всех } x \in X, & k = 0, 1, 2, \ldots; \\ p_k(x) > 0 & \text{для всех } x \notin X, & k = 0, 1, 2, \ldots; \\ p_k(x) \to \infty & \text{для всех } x \notin X, & k \to \infty; \\ p_{k+1}(x) > p_k(x) & \text{для всех } x \notin X, & k = 0, 1, 2, \ldots. \end{array}$$

Далее, используя алгоритмы глав 2 и 3, при каждом $k = 0, 1, 2, \ldots$ находим $\tilde{x}^k$ — решение задачи на безусловный минимум

$$f_0(x) + p_k(x) \to \min_{x \in R^n}. \qquad (5.21)$$

Предельные точки последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$ являются оптимальным решением задачи 1. Методы внешних штрафных функций обеспечивают приближенное решение задачи 1 вне допустимой области.

Приведем пример последовательности внешних штрафных функций для множества X, определенного согласно (5.20):

$$p_k(x) = \alpha_k \sum_{j=1}^{m} (\max\{0, f_j(x)\})^{\beta}, \quad k = 0, 1, 2, \ldots, \qquad (5.22)$$

где $\beta \geqslant 1$, $\{\alpha_k\}_{k=0}^{\infty}$ — строго возрастающая последовательность положительных чисел, стремящаяся к бесконечности при $k \to \infty$.

Алгоритм 1

I. Построить $p_k(x)$, $k = 0, 1, 2, \ldots$,— последовательность внешних штрафных функций для множества X, определенного согласно (5.20); для этого можно использовать (5.22).

II. При каждом $k = 0, 1, 2, \ldots$, найти $\tilde{x}^k$ — решение задачи на безусловный минимум

$$f_0(x) + p_k(x) \to \min_{x \in R^n} \qquad (5.23)$$

(для этого использовать алгоритмы, описанные в гл. 2 и 3).

III. Найти предельные точки последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$.

Теорема 1. *Пусть выполнены предположения 1. Тогда любая предельная точка последовательности* $\{\tilde{x}^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, является оптимальным решением задачи 1.*

Теорема 1'. *Пусть выполнены все предположения 1 и, кроме того:* а) *функции* f_j, $j = 0, 1, \ldots, m$ *— непрерывно дифференцируемы;* б) *множество*

$$X_\lambda = \{x \mid f_0(x) + p_k(x) \leqslant \lambda\}, \quad \lambda < \infty, \quad k = 0, 1, \ldots \qquad (5.24)$$

компактно; в) *задача 1 имеет единственное решение* x^*; г) *минимум в (5.23) при больших* k *достигается в единственной точке* $\tilde{x}^k$; д) *в решении* x^* *задачи 1 градиенты* $\nabla f_i(x^*)$, $i \in \mathcal{J}(x^*)$, *линейно независимы, где множество* $\mathcal{J}(x^*)$ *определяется по правилу*

$$\mathcal{J}(x^*) = \{j \mid f_j(x^*) = 0, \quad j = 1, \ldots, m\}.$$

Тогда для точек $\{\tilde{x}^k\}_{k=0}^{\infty}$, *порожденных алгоритмом 1, в котором* $p_k(x)$, $k = 0, 1, \ldots$, *определяется согласно* (5.22) *при* $\beta = 2$, *имеют*

место предельные соотношения

$$\lim_{k\to\infty} \tilde{x}^k = x^*;$$

$$\lim_{k\to\infty} \alpha_k (\max\{0,\ f_j(\tilde{x}^k)\}) = \frac{1}{2} u_j, \quad j = 1,\ \ldots,\ m,$$

где u_j, $j = 1, \ldots, m$ — множители Лагранжа задачи 1.

В случае задачи выпуклого программирования оценки приближения $\tilde{x}^k$ к искомому решению x^* могут быть сделаны более точными.

Теорема 1''. *Пусть функции f_j, $j = 0, 1, \ldots, m$, выпуклы и множество X_λ, определяемое согласно (5.24), компактно. Пусть, кроме того, в точке x^*, являющейся решением задачи 1, выполнены необходимые условия в форме теоремы Куна — Таккера, т. е. существуют такие числа $u_j \geqslant 0$, $j = 1, \ldots, m$, что*

$$f_0(x^*) \leqslant \sum_{j=1}^{m} u_j f_j(x) + f_0(x), \ \forall x;$$

$$u_j f_j(x^*) = 0, \quad j = 1,\ \ldots,\ m.$$

Тогда

$$f_j(\tilde{x}^k) \leqslant \frac{\bar{u}}{\alpha_k}, \text{ если } f_j(\tilde{x}^k) \geqslant 0;$$

$$f_0(\tilde{x}^k) \geqslant f_0(x^*) - \frac{5}{4}\frac{\bar{u}^2}{\alpha_k},$$

где $\bar{u} = \left(\sum_{j=1}^{m} (u_j)^2\right)^{1/2}$.

2. Задачи с ограничениями в виде равенств

Задача 2. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X = \{x \mid g_j(x) = 0, \quad j = 1,\ \ldots,\ m\}. \tag{5.25}$$

Предположения 2. (*i*) — функция f_0 непрерывно дифференцируема; (*ii*) — функции g_j, $j = 1, \ldots, m$, непрерывны; (*iii*) — существует такая точка x', что множество

$$X' = \{x \mid f_0(x) \leqslant f_0(x')\}$$

компактно.

Ниже приводится пример последовательности внешних штрафных функций для множества X, определенного согласно (5.25):

$$p_k(x) = \alpha_k \| g(x) \|^\beta = \alpha_k \left(\sum_{j=1}^{m} (g_j(x))^2\right)^{\beta/2}, \quad k = 0,\ 1,\ \ldots, \tag{5.26}$$

где $\beta \geqslant 1$ и α_k, $k = 0, 1, 2, \ldots$ — строго возрастающая последовательность положительных чисел, стремящаяся к бесконечности при $k \to \infty$.

Алгоритм 2

I. Построить $\{p_k(x)\}_{k=0}^{\infty}$ — последовательность внешних штрафных функций для множества X, определенного согласно (5.25); для этого можно использовать (5.26).

II. При каждом $k = 0, 1, \ldots$, найти $\tilde{x}^k$ — решение задачи на безусловный минимум

$$f_0(x) + p_k(x) \to \min_{x \in R^n}$$

(для этого использовать алгоритмы, описанные в гл. 2 и 3).

III. Найти предельные точки последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$.

Теорема 2. *Пусть выполнены предположения 2. Тогда любая предельная точка последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, является оптимальным решением задачи 2.*

3. Модифицированные методы с процедурой прерывания

Непосредственное использование описанных выше методов внешних штрафных функций не всегда практически осуществимо, так как не всегда возможно за конечное время найти глобальный минимум функции $f_0 + p_k$ на всем пространстве. Поэтому опишем модифицированные методы внешних штрафных функций, которые используют процедуру прерывания при приближении к минимуму или стационарным точкам функции $f_0 + p_k$. Последовательность точек $\tilde{x}^k$, $k = 0, 1, 2, \ldots$, порожденная этими алгоритмами, обладает свойством

$$\nabla f_0(\tilde{x}^k) + \nabla p_k(\tilde{x}^k) \to 0$$

при $k \to \infty$. При определенных условиях предельные точки этой последовательности удовлетворяют необходимым условиям оптимальности. Алгоритм 3 применяется для решения задачи 2, а алгоритм 3″ — для решения задачи 1.

Алгоритм 3

Начало. I. Выбрать числа $\varepsilon > 0$, $\alpha \in (0, 1/2)$, $\beta > 1$, $\rho > 0$ и точку $x^0 \in R^n$.

II. Положить $\varepsilon_0 = \varepsilon$, $j = 0$, $k = 0$.

Основной цикл. III. Вычислить вектор

$$h(x^j, \varepsilon_k) = -\left[\nabla f_0(x^j) + \frac{1}{\varepsilon_k} \nabla p(x^j)\right],$$

где функция p определяется по правилу

$$p(x) = \|g(x)\|^2/2 = \frac{1}{2} \sum_{l=1}^{m} (g_l(x))^2.$$

IV. Если $\|h(x^j, \varepsilon_k)\| > \varepsilon_k$, то перейти к шагу V; иначе положить $\varepsilon_{k+1} = \varepsilon_k/\beta$, $\tilde{x}^k = x^j$, $k = k + 1$ и перейти к шагу III.

V. Используя алгоритм 3′, вычислить такой шаговый множитель λ_j, что

$$-\lambda_j(1-\alpha)\|h(x^j, \varepsilon_k)\|^2 \leqslant f_0(x^j + \lambda_j h(x^j, \varepsilon_k)) + \frac{1}{\varepsilon_k} p(x^j + \lambda_j h(x^j, \varepsilon_k)) - f_0(x^j) - \frac{1}{\varepsilon_k} p(x^j) \leqslant -\lambda_j \alpha \|h(x^j, \varepsilon_k)\|^2.$$

VI. Положить $x^{j+1} = x^j + \lambda_j h(x^j, \varepsilon_k)$, $j = j + 1$ и перейти к шагу III.

Алгоритм 3′ **(алгоритм вычисления множителя λ_j для алгоритма 3)**

I. Положить $\mu = \rho$.

II. Вычислить

$$\underline{\theta}(\mu, x^j) = f_0(x^j + \mu h(x^j, \varepsilon_k)) + \frac{1}{\varepsilon_k} p(x^j + \mu h(x^j, \varepsilon_k)) - f_0(x^j) - \frac{1}{\varepsilon_k} p(x^j) + \mu(1-\alpha)\|h(x^j, \varepsilon_k)\|^2. \quad (5.27)$$

III. Если $\underline{\theta}(\mu, x^j) = 0$, то положить $\lambda_j = \mu$ и прекратить вычисления; если $\underline{\theta}(\mu, x^j) < 0$, то положить $\mu = \mu + \rho$ и перейти к шагу II; если $\underline{\theta}(\mu, x^j) > 0$, то перейти к шагу IV.

IV. Вычислить

$$\bar{\theta}(\mu, x^j) = f_0(x^j + \mu h(x^j, \varepsilon_k)) + \frac{1}{\varepsilon_k} p(x^j + \mu h(x^j, \varepsilon_k)) - f_0(x^j) - \frac{1}{\varepsilon_k} p(x^j) + \mu\alpha\|h(x^j, \varepsilon_k)\|^2. \quad (5.28)$$

V. Если $\bar{\theta}(\mu, x^j) \leqslant 0$, то положить $\lambda_j = \mu$ и прекратить вычисления; иначе положить $a_0 = \mu - \rho$, $b_0 = \mu$ и перейти к шагу VI.

VI. Положить $i = 0$.

VII. Положить $v_i = (a_i + b_i)/2$.

VIII. Вычислить $\underline{\theta}(v_i, x^j)$ и $\bar{\theta}(v_i, x^j)$, определяемые согласно (5.27) и (5.28), соответственно.

IX. Если $\underline{\theta}(v_i, x^j) \geqslant 0$ и $\bar{\theta}(v_i, x^j) \leqslant 0$, то положить $\lambda_j = v_i$ и прекратить вычисления; иначе перейти к шагу X.

X. Если $\underline{\theta}(v_i, x^j) > 0$, то положить $a_{i+1} = a_i$, $b_{i+1} = v_i$, $i = i + 1$ и перейти к шагу VII; иначе положить $a_{i+1} = v_i$, $b_{i+1} = b_i$, $i = i + 1$ и перейти к шагу VII.

Теорема 3. *Пусть выполнены предположения: (i) — функции f_0 и g_j, $j = 1, \ldots, m$, непрерывно дифференцируемы; (ii) — матрица Якоби $\frac{\partial g(x)}{\partial x}$ имеет максимальный ранг для всех x, принадлежа-*

щих достаточно большому открытому множеству в $\boldsymbol{R}^n$; (iii) — множество $\{x \mid f_0(x) \leqslant d_0\}$ компактно для всех $d_0 \in \boldsymbol{R}^1$.

Тогда, если последовательность $\{\tilde{x}^k\}_{k=0}^{\infty}$, порожденная алгоритмом 3, остается в компактном множестве, то $h(\tilde{x}^k, \varepsilon_k) \to 0$ при $k \to \infty$, последовательность $\{\tilde{x}^k\}_{k=0}^{\infty}$ будет иметь предельные точки x^*, которые лежат в X и удовлетворяют необходимым условиям оптимальности для задачи 2.

Замечание 3. Из теоремы 3 следует, что для эффективного использования «реализуемого» алгоритма 3 требуется выполнение ряда предположений, и в отличие от чисто принципиальной схемы (алгоритм 2) здесь, по-видимому, нет простых достаточно общих условий, гарантирующих, что последовательность $\{\tilde{x}^k\}_{k=0}^{\infty}$ будет оставаться в компактном множестве.

Алгоритм 3″

Шаги I, II, IV, V, VI такие, как в алгоритме 3. Шаг III алгоритма 3 заменить на III′.

III′. Вычислить вектор

$$h(x^j, \varepsilon_k) = -\left[\nabla f_0(x^j) + \frac{1}{\varepsilon_k}\nabla p(x^j)\right],$$

здесь функция

$$p(x) = \frac{1}{2}\sum_{l=1}^{m}(\max\{0, f_l(x)\})^2,$$

или

$$p(x) = \frac{1}{2}\sum_{l \in L(x)}[f_l(x)]^2,$$

где множество $L(x)$ определяется по правилу

$$L(x) = \{l \mid f_l(x) \geqslant 0, \quad l \in \{1, 2, \ldots, m\}\}.$$

Теорема 3′. *Пусть выполнены предположения: (i) — функции f_0 и f_l, $l = 1, \ldots, m$, непрерывно дифференцируемы;*

(ii) — множество $\{x \mid f_0(x) \leqslant d_0\}$ компактно для всех $d_0 \in \boldsymbol{R}^1$;

(iii) — для всех $x \in \boldsymbol{R}^n$ векторы $\nabla f_l(x)$, $l \in L(x)$, линейно независимы.

Тогда для последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3″, имеет место предельное соотношение $h(\tilde{x}^k, \varepsilon_k) \to 0$ при $k \to \infty$.

Если, кроме того, последовательность $\{\tilde{x}^k\}_{k=0}^{\infty}$ принадлежит компакту, то ее предельные точки x^* лежат в X и удовлетворяют необходимым условиям оптимальности для задачи 1.

Замечание 3′. Из теоремы 3′ следует, что для эффективного использования «реализуемого» алгоритма 3″ нет простых достаточно общих условий, гарантирующих сходимость последовательности

$\{\tilde{x}^k\}_{k=0}^{\infty}$. Однако на практике оказывается, что алгоритмы типа 3″ работают хорошо, поскольку обычно последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$ сходятся.

4. Метод внешних штрафных функций для минимизации недифференцируемых функций

Задача 4. Найти $\arg\min_{x \in X_1} f_0(x)$, где

$$X_1 = X \cap \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m\}. \tag{5.29}$$

Множество $X \subset R^n$ и функции $f_j : R^n \to R^1$ заданы.

Предположения 4. (*i*) — функции f_j, $j = 0, 1, \ldots, m$, выпуклые вниз и непрерывные при $x \in X$, (*ii*) — ограничения (5.29) удовлетворяют условию Слейтера; (*iii*) — X — выпуклое и замкнутое множество.

Приближенное решение данной задачи будем искать как решение вспомогательной (более «простой») задачи: найти $\arg\min_{x \in X} p(x, \alpha)$ для

$$p(x, \alpha) = f_0(x) + \sum_{j=1}^{m} \beta(f_j(x)) f_j(x), \tag{5.30}$$

где α — некоторое положительное число; $\beta(f_j(x)) = \alpha$, если $f_j(x) > 0$ и $\beta(f_j(x)) = 0$, если $f_j(x) \leqslant 0$.

Известно, что при α, превышающем абсолютные значения множителей Лагранжа, решение задачи (5.30) в точности совпадает с решением задачи 4.

Алгоритм 4

Начало. I. Выбрать: произвольное начальное приближение $x^0 \in R^n$; шаговый множитель ρ_0 и нормирующий множитель γ_0, удовлетворяющие условиям теоремы 4; положительное число α.

II. Положить $k = 0$.

Основной цикл. III. Вычислить числа $\beta_j = \beta(f_j(x^k))$, $j = 1, \ldots, m$.

IV. Вычислить следующее приближение

$$x^{k+1} = \pi_X\left(x^k - \rho_k \gamma_k \left(\hat{\nabla} f_0(x^k) + \sum_{j=1}^{m} \beta_j \hat{\nabla} f_j(x^k)\right)\right),$$

где $\hat{\nabla} f_j(x^k)$ $(j = 1, \ldots, m)$ — обобщенный градиент функции f_j в точке x^k.

V. Вычислить шаговый множитель ρ_{k+1} и нормирующий множитель γ_{k+1}, удовлетворяющие условиям теоремы 4.

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 4. Пусть выполнены предположения 4, а также:

(*i*) — *для любого числа* $\delta < \infty$ *найдется такое конечное число* $\chi = \chi(\delta)$, *что*

$$\|\hat{\nabla}_x p(x, \alpha)\| \leqslant \chi \quad \text{при} \quad \|x\| \leqslant \delta;$$

(*ii*) —

$$\gamma_k \geqslant 0, \quad \gamma_k \|\hat{\nabla}_x p(x^k, \alpha)\| \leqslant \text{const};$$

(*iii*) —

$$\rho_k \geqslant 0, \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty.$$

Тогда последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 4, сходится к решению задачи (5.30), а при достаточно большом* α — *к решению задачи 4.*

Замечание 4. Для ускорения сходимости функцию $p(x, \alpha)$ целесообразно минимизировать с помощью методов обобщенного градиентного спуска с растяжением пространства [223]. Чтобы получить гладкую функцию цели $p(x, \alpha)$ при гладких f_j, $j = 0, 1, \ldots, m$, вместо (5.30) можно взять функцию

$$p(x, \alpha) = f_0(x) + \sum_{j=1}^{m} \beta(f_j(x))[f_j(x)]^2. \tag{5.31}$$

Однако минимизация функции (5.31) дает лишь приближенное решение задачи 4, которое стремится к точному при $\alpha \to \infty$.

Замечания 4'. Методы внешних штрафных функций имеют следующие преимущества: а) в качестве начального приближения можно выбрать любую точку $x^0 \in \boldsymbol{R}^n$; б) их можно применять к задачам с ограничениями как типа равенств, так и типа неравенств; в) если штрафная функция имеет вид (5.22) и в процессе решения задачи минимизации без ограничений находим точку $\tilde{x}^k$, для которой $f_j(\tilde{x}^k) \leqslant 0$, то из суммы (в определении $p_k(x)$) исключается соответствующий член с функцией f_j и это уменьшает объем вычислений.

Недостатки методов внешних штрафных функций следующие: а) генерируют недопустимые, т. е. не всегда принадлежащие множеству X, точки; б) если минимизируемая функция $f_0(x)$ или некоторая из функций f_j, $j = 1, \ldots, m$ — невыпуклы, то при плохом начальном приближении может быть вычислен локальный, а не требуемый глобальный минимум функции $f_0 + p_k$ (в таких случаях требуются дополнительные ресурсы для поиска глобального минимума); в) штрафные функции вида (5.22) дифференцируемы небольшое число раз, что, согласно [373], может неблагоприятно сказываться на скорости сходимости; г) если вычислительный процесс начинается с внутренней точки x^0 множества X, то штрафная функция и все ее производные остаются равными нулю до тех пор, пока точка x^k не пересечет границу множества X (это делает вычислительный процесс неуправляемым внутри множества X).

Библиографические указания. Пункты 1, 2 написаны на основании работ [320, 285, 373, 464], пункт 3 — на основании работ [285, 536], пункт 4 — на основании работ [153, 294, 463].

В работе [427] для безусловной минимизации штрафной функции общей задачи нелинейного программирования используется модификация метода покоординатного спуска.

5.4. Методы внутренних штрафных функций

1. Общая схема

Задача 1. Найти $\arg\min\limits_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и для заданного множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j = 1, 2, \ldots, m, \quad x \in \boldsymbol{R}^n\}.$$

Предположения 1. (*i*) — функция f_0 непрерывно дифференцируема; (*ii*) — функции f_j, $j = 1, 2, \ldots, m$ — непрерывны; (*iii*) существует такая точка $x' \in X$, что множество

$$X' = \{x \mid f_0(x) \leqslant f_0(x')\}$$

компактно; (*iv*) — множество X имеет внутренние точки и замыкание его внутренности X^0 совпадает с X.

Идея метода заключается в сведении решения задачи с ограничениями к решению последовательности задач на безусловный минимум. При этом приходится строить последовательность внутренних штрафных функций для множества X, которая определяется следующим образом.

Определение. Последовательность непрерывных функций $p'_k : X^0 \to \boldsymbol{R}^1$, $k = 0, 1, 2 \ldots$, называется последовательностью внутренних штрафных функций для множества X^0, если:

$$0 < p'_{k+1}(x) < p'_k(x) \text{ для всех } x \in X^0, \quad k = 0, 1, 2, \ldots;$$

$$p'_k(x) \to 0 \text{ для всех } x \in X^0, \quad k \to \infty;$$

$p'_k(x^j) \to \infty$ при $j \to \infty$ для любой последовательности $\{x^j\}_{j=0}^{\infty}$, принадлежащей X^0, для которой $x^j \to x \in \partial X$ при $j \to \infty$, $k = 0, 1, 2, \ldots$ (здесь ∂X обозначает границу множества X).

Далее при каждом $k = 0, 1, 2, \ldots$, используя алгоритмы гл. 2 и 3, находят $\tilde{x}^k$ — решение задачи

$$f_0(x) + p'_k(x) \to \min_{x \in X^0}. \tag{5.32}$$

Предельные точки последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$ являются оптимальным решением задачи 1.

Ниже приведены примеры последовательностей внутренних штрафных функций для множества X:

$$p'_k(x) = -\alpha_k \sum_{j=1}^{m} \frac{1}{f_j(x)}, \quad k = 0, 1, 2, \ldots, \quad x \in X^0, \tag{5.33}$$

где функции f_j, $j = 1, 2, \ldots, m$, удовлетворяют предположению 1 и $\{\alpha_k\}_{k=0}^{\infty}$ — строго убывающая последовательность положительных чисел, стремящаяся к нулю при $k \to \infty$;

$$p'_k(x) = -\alpha_k \sum_{j=1}^{m} \log(-f_j(x)) =$$

$$= -\alpha_k \log((-1)^m f_1(x) f_2(x) \ldots f_m(x)), \; x \in X^0, \; k = 0, 1, \ldots, \tag{5.34}$$

где функции f_j, $j = 1, 2, \ldots, m$, и последовательность $\{\alpha_k\}_{k=0}^{\infty}$ такие, как в (5.33) и, кроме того, выполняется

$$\mu = \max_j (-\min_{x \in X} f_j(x)) < \infty.$$

Методы внутренних штрафных функций обеспечивают приближенное решение внутри допустимой области, в пределе стремящееся к решению задачи 1.

Алгоритм 1

I. Построить $p'_k(x)$, $k = 0, 1, 2, \ldots$ — последовательность внутренних штрафных функций к множеству X (можно использовать (5.33) и (5.34)).

II. При каждом $k = 0, 1, 2, \ldots$, найти $\tilde{x}^k$ — решение задачи минимизации функции $f_0(x) + p'_k(x)$ на множестве X^0 (можно использовать алгоритмы, описанные в гл. 2 и 3).

III. Найти предельные точки последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$.

Теорема 1. *Пусть выполнены предположения 1. Тогда любая предельная точка последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, является оптимальным решением задачи 1.*

2. Реализуемая схема с процедурой прерывания

Буквальное использование алгоритма 1 практически неосуществимо, так как невозможно за конечное время добиться минимизации функции $f_0(x) + p'_k(x)$ по $x \in X^0$. Ниже будет изложен модифицированный метод внутренних штрафных функций, использующий процедуры прерывания при решении задачи (5.32).

Алгоритм 2

Начало. I. Выбрать числа $\varepsilon > 0$, $\alpha \in (0, 1/2)$, $\beta > 1$, $\rho > 0$ и точку $x^0 \in X^0$.

II. Положить $\varepsilon_0 = \varepsilon$, $j = 0$, $k = 0$.

Основной цикл. III. Вычислить вектор

$$h(x^j, \varepsilon_k) = -(\nabla f_0(x^j) + \varepsilon_k \nabla p'(x^j)),$$

где функция $p'(x)$ такова, что $\varepsilon_k p'(x)$, $k = 0, 1, 2, \ldots$, — последовательность внутренних штрафных функций для множества X

($p'(x)$ можно определить, например, по правилам

$$p'(x) = -\sum_{l=1}^{m} \frac{1}{f_l(x)}, \quad x \in X^0,$$

или

$$p'(x) = -\sum_{l=1}^{m} \log(-f_l(x)), \quad x \in X^0).$$

IV. Если $\|h(x^j, \varepsilon_k)\| > \varepsilon_k$, то перейти к шагу V; иначе положить $\varepsilon_{k+1} = \varepsilon_k/\beta$, $\tilde{x}^k = x^j$, $k = k+1$ и перейти к шагу III.

V. Используя алгоритм 3' (см. § 5.3), вычислить такое число λ_j, что

$$x^j + \lambda_j h(x^j, \varepsilon_k) \in X$$

и

$$-\lambda_j(1-\alpha)\|h(x^j, \varepsilon_k)\|^2 \leqslant f_0(x^j + \lambda_j h(x^j, \varepsilon_k)) + \\ + \varepsilon_k p'(x^j + \lambda_j h(x^j, \varepsilon_k)) - f_0(x^j) - \varepsilon_k p'(x^j) \leqslant \\ \leqslant -\lambda_j \alpha \|h(x^j, \varepsilon_k)\|^2.$$

VI. Положить $x^{j+1} = x^j + \lambda_j h(x^j, \varepsilon_k)$, положить $j = j + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполнены предположения 1 и, кроме того: (i) — функции f_l, $l = 1, \ldots, m$ — непрерывно дифференцируемы; (ii) — для всех $x \in R^n$ векторы $\nabla f_l(x)$, $l \in L(x)$, линейно независимы, где $L(x) = \{l \mid f_l(x) \geqslant 0, l \in \overline{1, m}\}$; (iii) точка $x^0 \in X^0$ и число $\varepsilon > 0$ выбраны так, что множество $\{x \mid f_0(x) \leqslant f_0(x^0) + \varepsilon p'(x^0)\}$ компактно. Тогда предельные точки последовательности $\{\tilde{x}^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, допустимы и удовлетворяют необходимым условиям оптимальности задачи 1.*

Замечания 2. Достоинством методов внутренних штрафных функций (по сравнению с методами внешних штрафных функций) является то, что они порождают последовательности только допустимых точек, и, следовательно, могут быть использованы в реальном масштабе времени, имеют обычно производные достаточно высокого порядка и допускают управление вычислительным процессом во «внутренности» множества X.

Перечислим теперь их некоторые недостатки. Методы внутренних функций можно использовать лишь для ограниченного класса множеств X (для множеств X, содержащих внутренние точки). Процесс должен начинаться только из внутренней точки множества X. Если множество X имеет вид

$$X = \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m\},$$

то все члены штрафных функций типа (5.33) и (5.34) всегда дают ненулевой вклад в величину градиента штрафной функции. Это значительно увеличивает объем вычислений при отыскании направле-

ния спуска (градиента внутренней штрафной функции). В этом смысле методы внешних штрафных функций более предпочтительны.

3. Алгоритмы внутренней точки с применением Q-функций

Задача 3. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X = \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m, \quad x \in R^n\}.$$

Предположение 3. Функции f_j, $j = 0, 1, \ldots, m$, непрерывны, а множество

$$X^0 \triangleq \{x \mid f_i(x) > 0, \quad i = 1, \ldots, m\}$$

— непустое.

Главную роль в данном алгоритме играет характеристическое свойство функции, называемой *Q-функцией* и определенной следующим образом.

Определение. Пусть $z = (z_0, z_1, \ldots, z_m)$ обозначает $m + 1$-мерный положительный вектор. Функция $Q(z)$ называется Q-функцией если: а) $Q(z)$ непрерывна по $z > 0$; б) $\underline{\lim}_{i \to \infty} Q(z^i) = \alpha > Q(\bar{z})$, $\bar{z} > 0$ (допускается $\alpha = \infty$), причем нижний предел берется для такой бесконечной последовательности векторов $\{z^i\}$, что $z^i > 0$ для всех i и $\lim_{i \to \infty} \{z^i\} = \bar{z}$, причем $\bar{z}_j = 0$ для некоторого j.

В качестве примера Q-функции можно привести функцию

$$Q(f_0(x^k) - f_0(x), f_1(x), \ldots, f_m(x)) = \frac{1}{f_0(x^k) - f_0(x)} + \sum_{i=1}^{m} \frac{1}{f_i(x)}.$$

Алгоритмы внутренней точки с применением Q-функций составляют особый класс алгоритмов внутренней точки и обладают следующими свойствами:

1) не требуется выбирать строго убывающую последовательность $\{\alpha_i\}_{i=0}^{\infty}$ неотрицательных параметров, используемых в качестве весовых коэффициентов в случае обычной внутренней функции штрафа;

2) процесс минимизации на k-й итерации зависит только от значения целевой функции в начальной точке x^{k-1}, т. е. в точке безусловного минимума на $(k - 1)$-й итерации;

3) членам последовательности внутренних допустимых точек соответствуют значения целевой функции $f_0(x^0), f_0(x^1), \ldots, f_0(x^k), \ldots$ образующие, начиная с исходной точки, строго убывающую последовательность. При обычных предположениях эта последовательность локально сходится.

Алгоритм 3

Начало. I. Найти $x^0 \in X^0$.

II. Положить $k = 1$.

Основной цикл. III. Определить множество

$$X_k^0 = \{x \mid f_0(x) \leqslant f_0(x^{k-1}), \quad x \in X^0\}.$$

IV. Положить

$$Q^k(x) = Q(-f_0(x) + f_0(x^{k-1}), \quad f_1(x), \ \ldots, \ f_m(x)),$$

где $Q(\cdot)$ — Q-функция.

V. Найти x^k — какой-нибудь локальный минимум функции $Q^k(x)$ на множестве X_k^0.

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 3. *Пусть выполнены предположения 3 и: (i) множество X^* локальных минимумов задачи 3 — непустое компактное изолированное множество; (ii) $X^* \cap X^0 \neq \varnothing$.*

Тогда (i′) — при $X_k^0 \neq \varnothing$ существует точка x^k, являющаяся безусловным локальным минимумом функции Q^k на X_k^0, и любая предельная точка равномерно ограниченной последовательности $\{x^k\}_{k=0}^{\infty}$ есть локальный минимум функции f_0; (i′i′) — при $X_k^0 = \varnothing$ для некоторого конечного значения k точка x^{k-1} — локальный минимум функции f_0.

Замечания 3. Алгоритмы с Q-функцией сохраняют для задач выпуклого программирования все преимущества алгоритмов внутренних штрафных функций, например такие, как глобальная сходимость и возможность получения двойственных допустимых точек, дающих в пределе при $k \to \infty$ решения двойственной выпуклой задачи.

В [260] было показано, что минимизирующей последовательности для Q-функций соответствует минимизирующая последовательность для метода внутренних штрафных функций, отвечающая некоторой специальной последовательности значений коэффициентов штрафа.

Библиографические указания. При написании параграфа использовались работы [285, 536, 373, 464, 522, 528].

В работе [526] предлагается метод сопряженных внутренних штрафных функций, позволяющий преодолеть трудности, возникающие вблизи границы допустимой области в обычных методах внутренних штрафных функций.

5.5. Комбинированные методы штрафных функций

Задача 1. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества X.

Предположения 1. (*i*) — функция f_0 непрерывно дифференцируема; (*ii*) множество X имеет вид $X = X' \cap X''$, где X' замкнуто, а X'' замкнуто и замыкание его внутренности $X''^{,0}$ совпадает с X''; (*iii*) существует точка $\tilde{x} \in X$ такая, что множество $\tilde{X} = \{x \mid f_0(x) \leqslant$

$\leqslant f_0(\tilde{x})\}$ компактно; (*iv*) — по крайней мере для одной точки $\hat{x} \in X$, являющейся оптимальным решением задачи 1, найдется такой открытый шар V, что $\hat{x} \in V$ и пересечение внутренности множества X'' с множеством $V' = V \cap X'$ непусто.

Комбинированные методы штрафных функций представляют собой простую комбинацию методов внешних и внутренних штрафных функций. На каждой итерации приходится решать задачу минимизации функции без ограничений (минимизируется функция, являющаяся суммой целевой функции f_0, внутренней штрафной функции p'_k и внешней штрафной функции p_k). Предельные точки последовательности, состоящей из решений этих задач, при определенных условиях будут решениями задачи 1.

В общем случае комбинированные методы дают лучший результат вычисления условного минимума функций по сравнению с каждым методом в отдельности.

Алгоритм 1

I. Положить $k = 0$.

II. Построить функцию $p_k(x)$, принадлежающую последовательности внешних штрафных функций для множества X' (определение последовательности внешних штрафных функций и способы их построения приведены в § 5.3).

III. Построить функцию $p'_k(x)$, принадлежащую последовательности внутренних штрафных функций для множества X'' (определение последовательности внутренних штрафных функций и способы их построения приведены в § 5.4).

IV. Найти x^k — какой-нибудь локальный минимум задачи оптимизации без ограничений:

найти $\arg\min\limits_{x \in X'',0} (f_0(x) + p_k(x) + p'_k(x))$.

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Пусть выполнены предположения 1 и существует такая точка $x'' \in X''^{,0}$, что множество*

$$\{x \mid f_0(x) \leqslant f_0(x'') + p_0(x'') + p'_0(x'')\}$$

компактно. Тогда каждая предельная точка последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, является оптимальным решением задачи 1.

Библиографические указания. Параграф написан на основании работы [285]. В работе [530] предлагается комбинация метода порождающих столбцов и метода внешних штрафных функций, в работе [431] — комбинированный метод множителей и штрафных функций.

5.6. Стохастический метод штрафов

Задача 1. Найти $\arg\min_{x \in \mathscr{X}} EF_0(x, \omega)$, где

$$\mathscr{X} = X \cap \{x \mid EF_j(x, \omega) \leqslant 0, \quad j = 1, \ldots, m\}.$$

Предположения 1. (*i*) — значения $f_j(x) \triangleq EF_j(x, \omega)$, $j = 0, 1, \ldots$ $\ldots, m$, не вычисляются (или их практическое вычисление слишком трудоемко), а значения $F_j(x, \omega)$, $j = 0, 1, \ldots, m$ и градиенты этих функций вычисляются с требуемой точностью; (*ii*) — почти при каждом значении состояния природы ω функции $F_j(x, \omega)$, $j = 0, 1, \ldots$ $\ldots, m$, непрерывно дифференцируемы по x; (*iii*) — X — выпуклое замкнутое ограниченное множество.

Решение задачи 1 сводится к решению следующей задачи.

Задача 1′. Минимизировать по x функцию

$$p(x, c) = f_0(x) + \sum_{j=1}^{m} c_j(f_j(x)) f_j(x)$$

при ограничениях $x \in X$, где

$$c = c(\beta) = (c_1(\beta), \ldots, c_m(\beta));$$

$$c_j(\beta) = \begin{cases} \alpha, & \beta > 0, \\ 0, & \beta \leqslant 0, \quad j = 1, \ldots, m, \end{cases} \tag{5.34}$$

здесь α — достаточно большое число.

При достаточно большом конечном значении α решение задачи 1′ будет решением задачи 1.

Алгоритм 1

Начало. I. Выбрать: произвольные начальные приближения $x^0 \in R^n$ и $z^0 \in R^m$; начальные значения шаговых множителей $\rho_1(0)$ и $\rho_2(0)$, удовлетворяющие условиям теоремы 1; выбрать «достаточно большое» число $\alpha > 0$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить вектор ω^k (реализацию ω в k-й итерации).

IV. Вычислить вектор

$$x^{k+1} = \pi_X\left(x^k - \rho_1(k)\left[\nabla_x F_0(x^k, \omega^k) + \sum_{j=1}^{m} c_j(z_j^k) \nabla_x F_j(x^k, \omega^k)\right]\right),$$

где $c_j(z_j^k)$ — коэффициенты, определяемые согласно (5.34).

V. Вычислить вектор

$$z^{k+1} = \pi_Z(z^k + \rho_2(k)[F(x^k, \omega^k) - z^k]),$$

где $F = (F_1, \ldots, F_m)$, $Z \subset R^m$ — выпуклый компакт, содержащий векторы $\{(f_1(x), \ldots, f_m(x)), x \in X^*\}$ (здесь X^* — множество решений задачи 1).

VI. Найти значения шаговых множителей $\rho_1(k+1)$ и $\rho_2(k+1)$, удовлетворяющие условиям теоремы 1.

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполнены предположения 1 и, кроме того, имеют место условия*

$$\rho_1(k) \geqslant 0, \quad \rho_2(k) \geqslant 0 \quad \textit{при} \quad k = 0, 1, \ldots;$$

$$\sum_{k=0}^{\infty} \rho_1(k) = \infty, \quad \sum_{k=0}^{\infty} \rho_2(k) = \infty, \quad \frac{\rho_1(k)}{\rho_2(k)} \to 0 \quad \textit{при} \quad k \to \infty;$$

$$\sum_{k=0}^{\infty} E(\rho_1^2(k) + \rho_2^2(k)) < \infty.$$

Тогда каждая предельная точка последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, принадлежит множеству решений задачи 1′, а при достаточно большом значении α принадлежит множеству решений задачи 1.

В более общем случае стохастический метод штрафов имеет следующий вид.

Алгоритм 1′

Начало. I. Выбрать: произвольные начальные приближения $x^0 \in R^n$ и $z^0 \in R^m$; начальные значения шаговых множителей $\rho_1(0)$ и $\rho_2(0)$, удовлетворяющие условиям теоремы 1′; достаточно большое число $\alpha > 0$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализации $\xi^j(k)$, $j = 0, 1, \ldots, m$, случайных векторов ξ^j, $j = 0, 1, \ldots, m$, условное математическое ожидание которых

$$E(\xi^j/(x^0, z^0), \ldots, (x^k, z^k)) = \hat{\nabla} f_j(x^k) + b^j(k), \quad j = 0, 1, \ldots, m,$$

где $b^j(k)$, $j = 0, 1, \ldots, m$,— векторы, измеримые относительно σ-подалгебры, индуцированной случайными величинами $(x^0, z^0), \ldots$ $\ldots, (x^k, z^k)$.

IV. Вычислить вектор

$$x^{k+1} = \pi_X\left(x^k - \rho_1(k)\left[\xi^0(k) + \sum_{j=1}^{m} c_j(z_j^k)\xi^j(k)\right]\right),$$

где $c_j(z_j^k)$ — числа, определяемые при заданном z_j^k согласно (5.34).

V. Вычислить реализацию $\zeta(k)$ случайного вектора ζ, условное математическое ожидание которого

$$E(\zeta/(x^0, z^0), \ldots, (x^k, z^k)) = f(x^k),$$

где $f = (f_1, \ldots, f_m)$.

VI. Вычислить вектор

$$z^{k+1} = \pi_Z/(z^k + \rho_2(k)(\zeta(k) - z^k)).$$

VII. Вычислить значения шаговых множителей $\rho_1(k+1)$, $\rho_2(k+1)$, удовлетворяющие условиям теоремы 1′.

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1'. Пусть выполняются все условия теоремы 1 и, кроме того, имеет место неравенство

$$\sum_{k=0}^{\infty} \rho_1(k) \left\| \left(b^0(k) + \sum_{j=1}^{m} c_j(z_j^k) b^j(k) \right) \right\| < \infty \text{ п. н.}$$

Тогда каждая предельная точка последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1', принадлежит множеству решений задачи 1', а при достаточно большом значении* α *— множеству решений исходной задачи 1.*

Библиографические указания. При написании параграфа использовались работы [151, 153].

5.7. Методы возможных направлений

Вектор h^k называют возможным направлением в точке x^k (являющейся, например, k-м приближением к решению задачи отыскания $\arg\min_{x \in X} f_0(x)$), если существует такое число $\rho > 0$, что для всех $\lambda \in (0, \rho)$ точка $x^k + \lambda h^k$ является допустимой, т. е. $x^k + \lambda h^k \in X$ и, кроме того, удовлетворяет неравенству $f_0(x^k + \lambda h^k) < f_0(x^k)$.

В методах возможных направлений улучшенное $(k + 1)$-е приближение x^{k+1} к $\arg\min_{x \in X} f_0(x)$ вычисляют по формуле

$$x^{k+1} = x^k + \rho_k h^k.$$

Различные варианты методов возможных направлений отличаются способами выбора шаговых множителей ρ_k и возможных направлений h^k.

1. Методы возможных направлений решения задач минимизации с ограничениями типа неравенств

Задача 1. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j \in [1:m], \quad x \in R^n\},$$

определенного с помощью заданных функций $f_j : R^n \to R^1$, $j \in [1 : m]$.

Предположения 1. (*i*) — функции f_j, $j = 0, 1, \ldots, m$, непрерывно дифференцируемы; (*ii*) — множество X имеет внутренность или относительную внутренность (т. е. множество X может содержаться в линейном многообразии и иметь внутренность по отношению к этому многообразию).

Алгоритм 1

Начало. I. Вычислить начальное приближение $x^0 \in X$, удовлетворяющее условиям теоремы 1.

II. Положить $k = 0$.

Основной цикл. III. Положить $x = x^k$.

IV. Найти решение $\alpha = \alpha_k(x)$, $h = h^k(x)$ задачи линейного программирования в $(n+1)$-м пространстве векторов (α, h) : $\min\limits_{(\alpha,h)} \alpha$ при ограничениях

$$(\nabla f_0(x), h) \leqslant \alpha;$$
$$f_j(x) + (\nabla f_j(x), h) \leqslant \alpha, \quad j = 1, \ldots, m;$$
$$|h_i| \leqslant 1, \quad i = 1, \ldots, n.$$

V. Если $\alpha_k(x) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VI.

VI. Вычислить наименьшее положительное число ρ_k, удовлетворяющее условию

$$f_0(x + \rho_k h^k(x)) = \min_{\substack{\rho \geqslant 0 \\ x+\rho h^k(x) \in X}} f_0(x + \rho h^k(x)).$$

VII. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h^k(x).$$

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Если выполнены предположения 1 и начальное приближение $x^0 \in X$ таково, что множество*

$$X^0 \triangleq \{x \mid f_0(x) - f_0(x^0) \leqslant 0, \quad f_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}$$

компактно и имеет внутренность, то последовательность x^0, $x^1, \ldots$ $\ldots$, порожденная алгоритмом 1, либо конечна и ее последний элемент x^k удовлетворяет условию $\alpha_k(x^k) = 0$, либо бесконечна и каждая ее предельная точка $\bar{x}$ удовлетворяет условию $\alpha_k(\bar{x}) = 0$.

Если x^* — оптимальное решение задачи 1, то $\alpha(x^*) = 0$.

Основной недостаток алгоритма 1 состоит в том, что на каждой итерации требуется решать задачу линейного программирования, которая может иметь большую размерность (может включать все функции ограничений f_j, $j = 1, \ldots, m$).

В приводимом ниже алгоритме те из ограничений задачи линейного программирования

$$f_j(x) + (\nabla f_j(x), h) \leqslant \alpha, \quad j = 1, \ldots, m,$$

для которых значения $f_j(x)$ отрицательны и имеют достаточно большую абсолютную величину, могут оказаться несущественными и их исключают из задачи линейного программирования, на k-й итерации.

Алгоритм 1'

Начало. I. Вычислить начальное приближение $x^0 \in X$, удовлетворяющее условиям теоремы 1.

II. Выбрать константы $\varepsilon' > 0$, $\varepsilon'' \in (0, \varepsilon')$ и $\beta \in (0, 1)$ (рекомендуется $\beta = {}^1/_2$).

III. Положить $k = 0$.

IV. Положить $\varepsilon = \varepsilon'$.

Основной цикл. V. Положить $x = x^k$.

VI. Найти ε-активное множество индексов

$$\mathcal{J}_\varepsilon(x) \triangleq \{0\} \cup \{j \mid f_j(x) \geqslant -\varepsilon, \quad j \in [1:m]\}.$$

VII. Найти решение $\alpha = \alpha_\varepsilon(x)$, $h = h_\varepsilon^k$ задачи линейного программирования в $(n+1)$-мерном пространстве векторов (α, h) : $\min\limits_{(\alpha,h)} \alpha$ при ограничениях

$$(\nabla f_j(x), h) \leqslant \alpha, \quad j \in \mathcal{J}_\varepsilon(x);$$
$$|h_i| \leqslant 1, \quad i \in [1:n].$$

VIII. Если $\alpha_\varepsilon(x) \leqslant -\varepsilon$, то положить $h^k = h_\varepsilon^k$ и перейти к шагу XIII; иначе перейти к шагу IX.

IX. Если $\varepsilon \leqslant \varepsilon''$, то перейти к шагу X; иначе положить $\varepsilon = \beta\varepsilon$ и перейти к шагу VI.

X. Найти множество индексов

$$\mathcal{J}_0(x) \triangleq \{0\} \cup \{j \mid f_j(x) \geqslant 0, \quad j \in [1:m]\}.$$

XI. Найти решение $\alpha = \alpha_0(x)$, $h = h_0^k$ задачи линейного программирования в $(n+1)$-мерном пространстве векторов (α, h) : $\min\limits_{(\alpha,h)} \alpha$ при ограничениях

$$(\nabla f_j(x), h) \leqslant \alpha, \quad j \in \mathcal{J}_0(x);$$
$$|h_j| \leqslant 1, \quad j \in [1:n].$$

XII. Если $\alpha_0(x) = 0$, то положить $x^* = x$ и прекратить вычисления; иначе положить $\varepsilon = \beta\varepsilon$ и перейти к шагу VI.

XIII. Вычислить число

$$\bar\rho(x) = \max\{\rho \mid f_j(x + \lambda h^k) \leqslant 0, \quad \lambda \in [0, \rho], \quad j \in [1:m]\}.$$

XIV. Вычислить наименьшее значение $\rho_k \in [0, \bar\rho(x)]$, удовлетворяющее условию

$$f_0(x + \rho_k h^k) = \min_{\rho \in [0, \bar\rho(x)]} f_0(x + \rho h^k).$$

XV. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h^k.$$

XVI. Положить $k = k + 1$ и перейти к шагу V.

Для алгоритма 1′ имеет место теорема, аналогичная теореме 1.

В [285] рекомендуется заменить шаги XIII, XIV алгоритма 1′ практически реализуемыми шагами XIII′ и XIV′ для вычисления шагового множителя ρ_k.

XIII′. Вычислить наименьшее из целых чисел $s \geqslant 0$, для которых выполняются неравенства

$$f_0(x + \beta^s \delta h^k) - f_0(x) - \beta^s \delta \mu (\nabla f_0(x), h^k) \leqslant 0;$$
$$f_j(x + \beta^s \delta h^k) \leqslant 0, \quad j \in [1:m]$$

(здесь $\delta > 0$ и $\mu \in (0, 1)$ — наперед заданные константы).

XIV'. Положить $\rho_k = \beta^s \delta$.

Ниже приводится модификация алгоритма 1', в которой на каждой итерации необходимо решать более простую задачу линейного программирования.

Алгоритм 1''

Начало. I. Вычислить начальное приближение $x^0 \in X$, удовлетворяющее условиям теоремы 1'.

II. Выбрать константы $\varepsilon' > 0$, $\varepsilon'' \in (0, \varepsilon')$ и $\beta \in (0, 1)$ (рекомендуется $\beta = {}^1/_2$).

III. Положить $k = 0$.

Основной цикл. IV. Положить $x = x^k$.

V. Положить $\varepsilon_0 = \varepsilon'$.

VI. Положить $s = 0$.

VII. Вычислить ε_s — активное множество индексов

$$\mathcal{J}_{\varepsilon_s}(x) \triangleq \{j \mid f_j(x) \geqslant -\varepsilon_s, \quad j \in [1:m]\}.$$

VIII. Найти решение $h = h^k_{\varepsilon_s}$ задачи линейного программирования в n-мерном пространстве векторов h:

найти $\arg\min_h (\nabla f_0(x), h)$ при ограничениях

$$(\nabla f_j(x),\ h) \geqslant \varepsilon_s, \quad j \in \mathcal{J}_{\varepsilon_s}(x);$$
$$|h_i| \leqslant 1, \quad i \in [1:n].$$

IX. Вычислить значение

$$\alpha_{\varepsilon_s}(x) = (\nabla f_0(x),\ h^k_{\varepsilon_s}).$$

X. Если $\alpha_{\varepsilon_s}(x) \leqslant -\varepsilon_s$, то положить $h^k = h^k_{\varepsilon_s}$ и перейти к шагу XVI; иначе перейти к шагу XI.

XI. Если $\varepsilon_s \leqslant \varepsilon''$, то перейти к шагу XII; иначе положить $\varepsilon_{s+1} = \beta\varepsilon_s$, $s = s + 1$ и перейти к шагу VII.

XII. Вычислить множество индексов

$$\mathcal{J}_0(x) \triangleq \{j \mid f_j(x) \geqslant 0, \quad j \in [1:m]\}.$$

XIII. Найти решение $h = h^k_0$ задачи линейного программирования в n-мерном пространстве векторов h:

найти $\arg\min_h (\nabla f_0(x),\ h)$ при ограничениях

$$(\nabla f_j(x),\ h) \geqslant 0, \quad j \in \mathcal{J}_0(x);$$
$$|h_i| \leqslant 1, \quad i \in [1:n].$$

XIV. Вычислить значение $\alpha_0(x)$, $\alpha_0(x) = (\nabla f_0(x),\ h^k_0)$.

XV. Если $\alpha_0(x) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе положить $\varepsilon_{s+1} = \beta\varepsilon_s$, $s = s + 1$, и перейти к шагу VII.

XVI. Вычислить число $\bar{\rho}(x) \geqslant 0$:

$$\bar{\rho}(x) = \max\{\rho \mid f_j(x + \lambda h^k) \leqslant 0, \quad \lambda \in [0, \rho], \quad j \in [1:m]\}.$$

XVII. Вычислить наименьшее число $\rho_k \in [0, \bar{\rho}(x)]$, удовлетворяющее условию

$$f_0(x + \rho_k h^k) = \min_{\rho \in [0, \bar{\rho}(x)]} f_0(x + \rho h^k).$$

XVIII. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h^k.$$

XIX. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1'. *Если выполнены все предположения теоремы 1 и, кроме того, для любой точки $x \in X$ существует такой вектор $h \in \in \boldsymbol{R}^n$, что*

$$(\nabla f_j(x), h) < 0, \quad j \in \mathcal{J}_0(x),$$

где $\mathcal{J}_0(x)$ — *множество, определяемое соотношением*

$$\mathcal{J}_0(x) \triangleq \{j \mid f_j(x) \geqslant 0, \quad j \in [1:m]\},$$

то последовательность $x^0, x^1, \ldots$, порожденная алгоритмом 1'', либо конечна и ее последний элемент x^k удовлетворяет условию $\alpha_0(x^k) = 0$, либо бесконечна и каждая ее предельная точка $\bar{x}$ удовлетворяет условию $\alpha_0(\bar{x}) = 0$.

(Если x^* — оптимальное решение задачи 1, то $\alpha_0(x^*) = 0$).

2. Методы возможных направлений решения задач минимизации с ограничениями смешанного типа

Задача 2. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и множества X, заданного функциями $f_j : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $j = 1, \ldots, m$, матрицей A размера $l \times n$ и l-мерным вектором b,

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j \in [1:m], \quad Ax - b = 0, \quad x \in \boldsymbol{R}^n\}.$$

Предположения 2. (*i*) — функции f_j, $j = 0, 1, \ldots, m$ — выпуклы и непрерывно дифференцируемы; (*ii*) — множество X — компактно; (*iii*) — существует такая точка $\bar{x} \in \boldsymbol{R}^n$, что

$$A\bar{x} - b = 0, \quad f_j(\bar{x}) < 0, \quad j \in [1:m].$$

Алгоритм 2

Начало. I. Вычислить начальное приближение $x^0 \in X$.

II. Выбрать константы $\varepsilon_0 > 0$ и $\delta_j > 0$, $j = 0, 1, \ldots, m$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить множество индексов

$$\mathcal{J}_{\varepsilon_k}(x^k) \triangleq \{0\} \cup \{j \mid f_j(x^k) \geqslant -\varepsilon_k, \quad j \in [1:m]\}.$$

V. Найти решение $\alpha = \alpha_k$, $h = h^k$, задачи линейного программирования в $(n+1)$-мерном пространстве векторов $(\alpha, h) : \min_{(\alpha, h)} \alpha$ при ограничениях

$$(\nabla f_j(x^k), h) \leqslant \delta_j \alpha, \quad j \in \mathcal{J}_{\varepsilon_k}(x^k);$$

$$Ah = 0; \quad |h_i| \leqslant 1, \quad i = 1, \ldots, n.$$

VI. Если $\alpha_k = 0$, то перейти к шагу VII; иначе перейти к шагу X.

VII. Вычислить множество индексов

$$\mathcal{I}_0(x^k) \triangleq \{0\} \cup \{j \mid f_j(x^k) \geqslant 0, \quad j \in [1:m]\}.$$

VIII. Вычислить число

$$\gamma_k = -\max_{j \notin \mathcal{I}_0(x^k)} f_j(x^k).$$

IX. Если $\varepsilon_k < \gamma_k$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу X.

X. Если $\alpha_k < -\varepsilon_k$, то перейти к шагу XI; иначе перейти к шагу XVII.

XI. Положить $s = 0$.

XII. Положить $\rho_k = (^1/_2)^s$.

XIII. Если выполняются неравенства

$$f_0(x^k + \rho_k h^k) \leqslant f_0(x^k) + \frac{1}{2}\delta_0\rho_k\alpha_k;$$

$$f_j(x^k + \rho_k h^k) \leqslant 0, \quad j = 1, \ldots, m,$$

то перейти к шагу XV; иначе перейти к шагу XIV.

XIV. Положить $s = s + 1$ и перейти к шагу XII.

XV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

XVI. Положить $\varepsilon_{k+1} = \varepsilon_k$, $k = k + 1$ и перейти к шагу IV.

XVII. Положить $x^{k+1} = x^k$, $\varepsilon_{k+1} = \varepsilon_k/2$, $k = k + 1$, и перейти к шагу IV.

Теорема 2. *Если выполнены предположения 2 и, кроме того, градиенты функций f_j, $j = 1, \ldots, m$, удовлетворяют условию Липшица*

$$\|\nabla f_j(x) - \nabla f_j(y)\| \leqslant \gamma \|x - y\|, \quad \gamma < \infty, \ x, \quad y \in \boldsymbol{R}^n,$$

то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, обладает тем свойством, что $\{f_0(x^k)\}_{k=0}^{\infty}$ монотонно не возрастая, стремится к значению $f_0(x^*)$, где x^* — точка минимума функции $f_0(x)$ в области X. Если x^* — единственная точка минимума, то $x^k \to x^*$ при $k \to \infty$.

Ниже приводятся рекомендуемые в [285] два метода возможных направлений, удобные для реализации их на ЭВМ.

Алгоритм 2'

Начало. I. Выбрать константы $\varepsilon' > 0$, $\varepsilon'' \in (0, \varepsilon')$, $\lambda > 0$, $\beta' \in (0, 1)$, $\beta'' \in (0{,}5; 0{,}8)$ и целое число τ, удовлетворяющее условию $5 \leqslant \tau \leqslant 10$.

II. Если имеется вектор x', удовлетворяющий условиям

$$f_j(x') \leqslant 0, \quad j \in [1:m];$$
$$Ax' = b,$$

то положить $x^0 = x'$ и перейти к шагу VII; иначе перейти к шагу III.

III. Вычислить вектор $\bar{x}^0$, удовлетворяющий условию $A\bar{x}^0 = b$.

IV. Вычислить

$$\alpha_0 = \max\{f_j(\bar{x}^0), \quad j \in [1:m]\}.$$

V. Использовать шаги VII—XXI алгоритма 2' для решения задачи минимизации в $(n+1)$-мерном пространстве векторов $(\alpha, \bar{x})$: $\min\limits_{(\alpha, \bar{x})} \alpha$ при ограничениях

$$f_j(\bar{x}) \leqslant \alpha, \quad j \in [1:m];$$
$$A\bar{x} = b$$

с начальным приближением $(\alpha_0, \bar{x}^0)$ до тех пор, пока при некотором $k = k_0$ не окажется, что

$$f_j(\bar{x}^{k_0}) \leqslant 0, \quad j \in [1:m], \quad A\bar{x}^{k_0} = 0.$$

VI. Положить $x^0 = \bar{x}^{k_0}$ и перейти к шагу VII.

Основной цикл. VII. Положить $k = 0$.

VIII. Положить $\varepsilon = \varepsilon'$.

IX. Положить $x = x^k$.

X. Найти множества индексов $\mathcal{J}_\varepsilon^1$, $\mathcal{J}_\varepsilon^2$, определяемые соотношениями

$$\mathcal{J}_\varepsilon^1 \cup \mathcal{J}_\varepsilon^2 = \{0\} \cup \{j \mid f_j(x) + \varepsilon \geqslant 0, \quad j \in [1:m]\};$$
$$\mathcal{J}_\varepsilon^1 \cap \mathcal{J}_\varepsilon^2 = \varnothing;$$
$$0 \in \mathcal{J}_\varepsilon^2;$$

$j \in \mathcal{J}_\varepsilon^1$, только если функция f_j аффинна.

XI. Решить задачу линейного программирования в $(n+1)$-мерном пространстве векторов (α, h) :

$\min\limits_{(\alpha, h)} \alpha$ при ограничениях

$$(\nabla f_j(x), h) \leqslant \alpha, \quad j \in \mathcal{J}_\varepsilon^2;$$
$$(\nabla f_j(x), h) \leqslant 0, \quad j \in \mathcal{J}_\varepsilon^1;$$
$$|h_i| \leqslant 1, \quad i = 1, \ldots, n;$$
$$Ah = 0$$

и обозначить ее решение через $(\alpha_\varepsilon^0, h_\varepsilon^k)$.

XII. Если $\alpha_\varepsilon^0 \leqslant -\lambda\varepsilon$, то положить $h^k = h_\varepsilon^k$ и перейти к шагу XVI; иначе перейти к шагу XIII.

XIII. Если $\varepsilon \leqslant \varepsilon''$, то перейти к шагу XIV; иначе положить $\varepsilon = \beta'\varepsilon$ и перейти к шагу X.

XIV. Положить $\bar{\varepsilon} = \varepsilon$.

XV. Положить $\varepsilon = 0$ и использовать шаги X, XI алгоритма 2′ для вычисления вектора (α_0^0, h_0^k). Если $\alpha_0^0 = 0$, то положить $x^* = x$ и прекратить вычисления; иначе положить $\varepsilon = \beta'\bar{\varepsilon}$ и перейти к шагу X.

XVI. Положить $s = 0$.

XVII. Положить $\rho_k = (\beta'')^s$.

XVIII. Если выполняются неравенства

$$f_0(x + \rho_k h^k) - f_0(x) - \frac{1}{2}\rho_k(\nabla f_0(x),\ h^k) \leqslant 0;$$

$$f_j(x + \rho_k h^k) \leqslant 0, \quad j \in [1:m],$$

то перейти к шагу XIX; иначе положить $s = s + 1$ и перейти к шагу XVII.

XIX. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h^k.$$

XX. Положить $k = k + 1$.

XXI. Если число k кратно τ, то перейти к шагу VIII; иначе перейти к шагу IX.

Алгоритм 2″ (алгоритм 2″ можно применять для решения задачи 2 лишь при выполнении условий регулярности Куна — Таккера)

Начало. I. Выбрать константы $\varepsilon' > 0$, $\varepsilon'' \in (0, \varepsilon')$, $\lambda' > 0$, $\lambda'' > 0$, $\beta' \in (0, 1)$, $\beta'' \in (0{,}5;\ 0{,}8)$ и целое число τ, удовлетворяющее условию $5 \leqslant \tau \leqslant 10$.

II. Вычислить начальное приближение x^0, удовлетворяющее условиям (шаги II—VI алгоритма 2′)

$$f_j(x^0) \leqslant 0, \quad j \in [1:m];$$

$$Ax^0 = b.$$

III. Положить $k = 0$.

Основной цикл. IV. Положить $\varepsilon = \varepsilon'$.

V. Положить $x = x^k$.

VI. Найти множества индексов $\mathcal{I}_\varepsilon^1$ и $\mathcal{I}_\varepsilon^2$, как на шаге X алгоритма 2′.

VII. Решить задачу линейного программирования в n-мерном пространстве векторов h: минимизировать $(\nabla f_0(x), h)$ при ограничениях

$$(\nabla f_j(x),\ h) + \lambda'\varepsilon \leqslant 0, \quad j \neq 0, \quad j \in \mathcal{I}_\varepsilon^2;$$

$$(\nabla f_j(x),\ h) \leqslant 0, \quad j \in \mathcal{I}_\varepsilon^1;$$

$$|h_i| \leqslant 1, \quad i = 1, \ldots, n;$$

$$Ah = 0$$

и обозначить ее решение через h_ε^k.

VIII. Вычислить

$$\alpha_\varepsilon = (\nabla f_0(x),\ h_\varepsilon^k).$$

IX. Если $\alpha_\varepsilon \leqslant -\lambda''\varepsilon$, то положить $h^k = h_\varepsilon^k$ и перейти к шагу XV; иначе перейти к шагу X.

X. Если $\varepsilon \leqslant \varepsilon''$, то перейти к шагу XI; иначе положить $\varepsilon = \beta'\varepsilon$ и перейти к шагу VI.

XI. Положить $\bar{\varepsilon} = \varepsilon$.

XII. Положить $\varepsilon = 0$ и использовать шаги VI, VII алгоритма 2″ для вычисления вектора h_0^k.

XIII. Вычислить

$$\alpha_0 = (\nabla f_0(x),\ h_0^k).$$

XIV. Если $\alpha_0 = 0$, то положить $x^* = x$ и прекратить вычисления; иначе положить $\varepsilon = \beta'\bar{\varepsilon}$ и перейти к шагу VI.

XV. Положить $s = 0$.

XVI. Положить $\rho_k = (\beta'')^s$.

XVII. Если выполняются неравенства

$$f_0(x + \rho_k h^k) - f_0(x) - \frac{1}{2}\rho_k(\nabla f_0(x),\ h^k) \leqslant 0;$$

$$f_j(x + \rho_k h^k) \leqslant 0, \quad j \in [1:m],$$

то перейти к шагу XVIII; иначе положить $s = s + 1$ и перейти к шагу XVI.

XVIII. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h^k.$$

XIX. Положить $k = k + 1$.

XX. Если число k кратное τ, то перейти к шагу IV; иначе перейти к шагу V.

3. Метод возможных направлений с квадратичным поиском

В методе возможных направлений с квадратичным поиском для вычисления вектора движения h^k к улучшенному $(k + 1)$-му приближению к искомому решению используются члены, содержащие вторые производные минимизируемой функции. Очевидно, эти алгоритмы имеют большую скорость сходимости, чем алгоритмы с линейным поиском, однако на каждой итерации требуется решать задачу квадратичного программирования. Предложенный в [285] алгоритм 3 можно применять для решения задачи 2 в том случае, когда собственные значения матрицы Гессе $H(x) \triangleq \partial^2 f_0(x)/\partial x^2$ далеко разнесены, когда выполнены условия регулярности Куна — Таккера и матрица вторых производных $H(x) \triangleq \partial^2 f_0(x)/\partial x^2$ положительно полуопределена на множестве

$$X_0 \triangleq \{x \mid f_0(x) \leqslant f_0(x^0),\quad f_j(x) \leqslant 0,\quad j \in [1:m],\quad Ax = b\},$$

где x^0 — начальное приближение.

Алгоритм 3

Начало. I. Выбрать константы $\varepsilon' > 0$, $\varepsilon'' \in (0, \varepsilon')$, $\lambda' > 0$, $\lambda'' > 0$, $\beta' \in (0, 1)$, $\beta'' \in (0{,}5; 0{,}8)$ и целое число τ, удовлетворяющее условию $5 \leqslant \tau \leqslant 10$.

II. Вычислить начальное приближение x^0, удовлетворяющее условиям

$$f_j(x^0) \leqslant 0, \quad j \in [1:m], \quad Ax^0 = b$$

(о методе вычисления x^0 см. шаги II—VI алгоритма 2′).

III. Положить $k = 0$.

Основной цикл. IV. Положить $\varepsilon = \varepsilon'$.

V. Положить $x = x^k$.

VI. Найти множества индексов $\mathcal{I}_\varepsilon^1$, $\mathcal{I}_\varepsilon^2$, как на шаге X алгоритма 2′.

VII. Найти решение $h = h_\varepsilon^k$ задачи квадратичного программирования в n-мерном пространстве:

$$\text{найти } \arg\min_h \left(\frac{1}{2} \left(\frac{\partial^2 f_0(x)}{\partial x^2} h,\ h \right) + (\nabla f_0(x),\ h) \right)$$

при ограничениях

$$(\nabla f_j(x),\ h) + \lambda' \varepsilon \leqslant 0, \quad j \neq 0, \quad j \in \mathcal{I}_\varepsilon^2;$$

$$(\nabla f_j(x),\ h) \leqslant 0, \quad j \in \mathcal{I}_\varepsilon^1;$$

$$|h_i| \leqslant 1, \quad i = 1, \ldots, n;$$

$$Ah = b.$$

VIII. Вычислить

$$\alpha_\varepsilon = \frac{1}{2} \left(\frac{\partial^2 f_0(x)}{\partial x^2} h_\varepsilon^k,\ h_\varepsilon^k \right) + (\nabla f_0(x),\ h_\varepsilon^k).$$

IX. Если $\alpha_\varepsilon \leqslant -\lambda'' \varepsilon$, то положить $h^k = h_\varepsilon^k$ и перейти к шагу XIV; иначе перейти к шагу X.

X. Если $\varepsilon \leqslant \varepsilon''$, то перейти к шагу XI; иначе положить $\varepsilon = \beta' \varepsilon$ и перейти к шагу VI.

XI. Положить $\bar{\varepsilon} = \varepsilon$.

XII. Положить $\varepsilon = 0$ и использовать шаги VI—VIII алгоритма 3 для вычисления значения

$$\alpha_0 = \frac{1}{2} \left(\frac{\partial^2 f_0(x)}{\partial x^2} h_0^k,\ h_0^k \right) + (\nabla f_0(x),\ h_0^k).$$

XIII. Если $\alpha_0 = 0$, то положить $x^* = x$ и прекратить вычисления; иначе положить $\varepsilon = \beta' \bar{\varepsilon}$ и перейти к шагу VI.

XIV. Положить $s = 0$.

XV. Положить $\rho_k = (\beta'')^s$.

XVI. Если выполняются неравенства

$$f_0(x+\rho_k h^k)-f_0(x)-\frac{1}{2}\rho_k(\nabla f_0(x),\ h^k)\leqslant 0;$$

$$f_j(x+\rho_k h^k)\leqslant 0,\quad j\in[1:m],$$

то перейти к шагу XVII; иначе положить $s=s+1$ и перейти к шагу XV.

XVII. Вычислить следующее приближение

$$x^{k+1}=x+\rho_k h^k.$$

XVIII. Положить $k=k+1$.

XIX. Если число k кратное τ, то перейти к шагу IV; иначе перейти к шагу V.

4. Модифицированный метод возможных направлений Зойтендейка

Задача 4. Найти $\arg\min\limits_{x\in X} f_0(x)$ для заданной непрерывной функции $f_0: \boldsymbol{R}^n\to\boldsymbol{R}^1$ и множества

$$X\triangleq\{x\,|\,f_j(x)=0,\quad j=1,\ \ldots,\ l;\quad f_j(x)\geqslant 0,$$
$$j=l+1,\ \ldots,\ m;\quad x\in\boldsymbol{R}^n\},$$

где $f_j: \boldsymbol{R}^n\to\boldsymbol{R}^1$, $j=1,\ \ldots,\ m$ — заданные непрерывные функции.

Предположения 4. (*i*) — функции $f_j(x)$, $j=0,\ 1,\ \ldots,\ m$ непрерывно дифференцируемы в $\boldsymbol{R}^n$; (*ii*) — $l<n$.

Решение задачи 4 сводится к решению следующей вспомогательной задачи:

$$\text{найти } \arg\min_x\left(f_0(x)+\alpha\sum_{j=1}^{l}f_j(x)\right) \text{ при ограничениях}$$

$$-f_j(x)\leqslant 0,\quad j=1,\ \ldots,\ m,$$

где $\alpha>0$ — некоторый параметр.

Любой минимум вспомогательной задачи, локальный или глобальный, достигаемый в точке $\hat{x}$ с $f_j(\hat{x})=0$, $j=1,\ \ldots,\ l$, является также локальным или глобальным минимумом задачи 4. Для решения вспомогательной задачи используется метод возможных направлений Зойтендейка, дополненный процедурой вычисления подходящего значения параметра α.

Алгоритм 4

Начало. I. Найти начальное приближение $x^0\in X^+$, где

$$X^+=\{x\,|\,f_j(x)\geqslant 0,\quad j=1,\ \ldots,\ m,\quad x\in\boldsymbol{R}^n\}. \tag{5.35}$$

II. Выбрать константы $\beta>0$, $\varepsilon_0>0$, $\varepsilon'>0$ $(\varepsilon'\ll\varepsilon_0)$, $\eta\in(0,\ 1)$, $\theta\in(0,\ 1)$, $\alpha_{-1}>0$, $\delta>0$.

III. Для заданных чисел $\alpha > 0$, $\varepsilon > 0$ определить функции ψ_α: $\boldsymbol{R}^n \to \boldsymbol{R}^1$ и $\mathcal{I}_\varepsilon$: $\boldsymbol{R}^n \to \{1, \ldots, m\}$ по правилам

$$\psi_\alpha(x) \triangleq f_0(x) + \alpha \sum_{j=1}^{l} f_j(x);$$

$$\mathcal{I}_\varepsilon(x) \triangleq \{j \mid -f_j(x) \geqslant -\varepsilon, \quad j \in [1:m]\}.$$

IV. Определить функцию $g_{\alpha\varepsilon}$: $\boldsymbol{R}^n \to \boldsymbol{R}^1$ по правилу

$$g_{\alpha\varepsilon}(x) = \min_{h \in S} \max \{(\nabla\psi_\alpha(x), h), \ -(\nabla f_j(x), h), \quad j \in \mathcal{I}_\varepsilon(x)\}, \quad (5.36)$$

где $S = \{h \mid |h_i| \leqslant 1, \quad i = 1, \ldots, n, \quad h \in \boldsymbol{R}^n\}$.

V. Положить $k = 0$, $\varepsilon = \varepsilon_0$.

Основной цикл. VI. Вычислить множители Лагранжа $\lambda(x^k) = (\lambda_1(x^k), \ldots, \lambda_m(x^k))$ по правилу

$$\lambda(x^k) = [\nabla f(x^k)(\nabla f(x^k))^T]^{-1} [\nabla f(x^k)] \nabla f_0(x^k),$$

где $f(x^k) = (f_1(x^k), \ldots, f_m(x^k))^T$; $\nabla f(x^k)$ — соответствующая $m \times$ $\times\, n$-матрица.

VII. Вычислить

$$\lambda_0 = \min\{\lambda_j(x^k), \quad j \in [1:l]\}.$$

VIII. Если $\alpha_{k-1} + \lambda_0 \geqslant \beta$, то положить $\alpha_k = \alpha_{k-1}$ и перейти к шагу IX; если $\alpha_{k-1} + \lambda_0 < \beta$, то положить $\alpha_k = \max\{\beta - \lambda_0,$ $\alpha_{k-1} + \delta\}$ и перейти к шагу IX.

IX. Вычислить множество индексов $\mathcal{I}_\varepsilon(x^k)$ и векторы

$$\nabla\psi_{\alpha_k}(x^k), \quad \nabla f_j(x^k), \quad j \in \mathcal{I}_\varepsilon(x^k).$$

X. Вычислить значение $g_{\alpha_k\varepsilon}(x^k)$ и найти возможное направление поиска $h(x^k)$, решая следующую задачу

$$\text{найти } \arg\min_{h \in S} \max\{(\nabla\psi_{\alpha_k}(x^k), h); \ -(\nabla f_j(x^k), h), \ j \in \mathcal{I}_\varepsilon(x^k)\},$$

где $S = \{h \mid |h_i| \leqslant 1, \quad i = 1, \ldots, n, \quad h \in \boldsymbol{R}^n\}$.

XI. Если $g_{\alpha_k\varepsilon}(x^k) > -\varepsilon$, то положить $\varepsilon = \varepsilon/2$ и перейти к шагу XII; иначе перейти к шагу XIII.

XII. Если $\varepsilon < \varepsilon'$, то вычислить значение $g_{\alpha_k 0}(x^k)$ и прекратить вычисления, если $g_{\alpha_k 0}(x^k) = 0$; иначе перейти к шагу IX.

XIII. Вычислить наименьшее целое $i_k \geqslant 0$ такое, что

$$f_j(x^k + \eta^{i_k} h(x^k)) \geqslant 0, \quad j = 1, \ldots, m;$$

$$\psi_{\alpha_k}(x^k + \eta^{i_k} h(x^k)) - \psi_{\alpha_k}(x^k) \leqslant -\theta\eta^{i_k}\varepsilon.$$

XIV. Положить $x^{k+1} = x^k + \eta^{i_k} h(x^k)$, $k = k + 1$ и перейти к шагу VI.

Теорема 4. *Пусть выполняются предположения 4 и* (***iii***) *— для любого $x \in \boldsymbol{R}^n$ число линейно независимых векторов $\nabla f_j(x)$, $j \in [1:m]$ таких, что $f_j(x) = 0$ не превышает n;* (***iv***) *— множество*

$$X^+ \triangleq \{x \mid f_j(x) \geqslant 0, \quad j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}$$

имеет непустую внутренность, замыкание которой совпадает с X^+.

Тогда:

1) *если алгоритм 4 останавливается в точке x^ν, то*

$$g_{\alpha_\nu 0}(x^\nu) = 0 \quad \text{и} \quad \sum_{j=1}^{l} f_j(x^\nu) = 0;$$

2) *если алгоритм 4 генерирует бесконечную последовательность $\{x^k\}_{k=0}^{\infty}$ с возрастающими значениями α_k, $k = k_1, k_2, \ldots, k_t$ так, что после k_t значения α_k сохраняются постоянными и равными $\bar{\alpha}$, то любая точка накопления x^* последовательности $\{x^k\}_{k=0}^{\infty}$ с $\mathcal{J}_\varepsilon(x^*) \equiv \mathcal{J}_0(x^*)$ удовлетворяет условию*

$$g_{\bar{\alpha}0}(x^*) = 0, \quad \sum_{j=1}^{l} f_j(x^*) = 0;$$

3) *если алгоритм 4 генерирует бесконечную последовательность $\{x^k\}_{k=0}^{\infty}$, для которой значения α_k возрастают бесконечно часто при $k = k_1, k_2, \ldots$, то последовательность $\{x^{k_i}\}_{i=1}^{\infty}$ не имеет точки накопления.*

Теорема 4'. *Пусть выполняются все условия теоремы 4, и пусть $\tilde{X}$ — компактное подмножество в X^+. Тогда существует число $\tilde{\alpha} \in (0, \infty)$ такое, что если $\hat{x} \in \tilde{X}$ точка локального (глобального) минимума следующей задачи:*
найти

$$\arg\min_{x \in X^+} \left(f_0(x) + \alpha \sum_{j=1}^{l} f_j(x) \right), \quad \alpha > \tilde{\alpha}, \tag{5.37}$$

то $\hat{x}$ есть также точка локального (глобального) минимума задачи 4.

Теорема 4''. *Пусть выполняются все условия теоремы 4. Тогда если $\hat{x}$ решение задачи (5.37), то $g_{\alpha 0}(\hat{x}) = 0$, где $g_{\alpha 0}(\hat{x})$ определяется по (5.36) при $\varepsilon = 0$.*

5. Аналог метода возможных направлений в задачах минимизации почти дифференцируемых функций

З а д а ч а 5. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного множества ограничений

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}. \tag{5.38}$$

Предположения 5. (*i*) — f_0— почти дифференцируемая функция; (*ii*) — функции f_j, $j = 1, \ldots, m$ непрерывно дифференцируемые.

Алгоритм 5

Начало. I. Выбрать: начальное приближение $x^0 \in X$, произвольный вектор $z^0 \in R^n$; положить $k = 0$.

Основной цикл. II. Найти числа ρ_k'', γ_k, α_k, ε_k, удовлетворяющие условиям теоремы 5.

III. Найти множество индексов $\mathcal{J}(x^k, \varepsilon^k)$, определяемое соотношением

$$\mathcal{J}(x^k, \varepsilon_k) = \{j \mid -\varepsilon_k \leqslant f_j(x^k) \leqslant 0, \quad j = 1, \ldots, m\}.$$

IV. Вычислить $\tilde{x}_i^k$, $i = 1, \ldots, n$ — независимые реализации случайных величин, равномерно распределенных на отрезках

$$\left[x_i^k - \frac{\alpha_k}{2}, \; x_i^k + \frac{\alpha_k}{2}\right], \quad i = 1, \ldots, n.$$

V. Вычислить вектор

$$\theta(x^k, k) = \frac{1}{\alpha_k} \sum_{i=1}^{n} \left[f_0\left(\tilde{x}_1^k, \ldots, x_i^k + \frac{\alpha_k}{2}, \ldots, \tilde{x}_n^k\right) - \right.$$
$$\left. - f_0\left(\tilde{x}_1^k, \ldots, x_i^k - \frac{\alpha_k}{2}, \ldots, \tilde{x}_n^k\right)\right] e^i, \tag{5.39}$$

где e^i, $i = 1, \ldots, n$ — i-й орт.

VI. Найти вектор $h = h^k$ — решение следующей задачи линейного программирования:
найти $\max\limits_{h, \beta} \beta$ при ограничениях

$$(z^k, h) + \beta \leqslant 0;$$
$$(\nabla f_j(x^k), h) + \beta \leqslant 0, \quad j \in \mathcal{J}(x^k, \varepsilon_k);$$
$$-1 \leqslant h_j \leqslant 1, \quad j = 1, \ldots, n.$$

VII. Найти число ρ_k' из условия

$$\rho_k' = \max\{\rho \mid x^k + \rho h^k \in X\}.$$

VIII. Вычислить шаговый множитель

$$\rho_k = \min\{\rho_k', \rho_k''\}.$$

IX. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

X. Вычислить вектор

$$z^{k+1} = z^k + \gamma_k(\theta(x^k, k) - z^k).$$

XI. Положить $k = k + 1$ и перейти к шагу II.

Теорема 5. *Пусть выполнены предположения 5 и* (*iii*) — *область* X, *высекаемая ограничениями* (5.38), *ограничена;* (*iv*) — *для за-*

дачи 5 выполняются условия Слейтера; (v) — числа ε_k, ρ_k'', γ_k, α_k, $k = 0, 1, \ldots$, удовлетворяют следующим условиям:

$$\varepsilon_k > 0, \quad \alpha_k > 0, \quad \rho_k'' > 0, \quad \gamma_k > 0, \quad k = 0, 1, \ldots;$$

$$\sum_{k=0}^{\infty} \rho_k'' = \infty, \quad \sum_{k=0}^{\infty} (\rho_k''/\alpha_k)^2 < \infty;$$

$$\sum_{k=0}^{\infty} \gamma_k^2 < \infty; \quad \rho_k''/\alpha_k\gamma_k \to 0 \text{ при } k \to \infty;$$

$$\frac{|\alpha_k - \alpha_{k+1}|}{\rho_k''} \to 0 \text{ при } k \to \infty; \quad \alpha_k \to 0 \text{ при } k \to \infty;$$

$$\varepsilon_k \to 0 \text{ при } k \to \infty.$$

Тогда с вероятностью 1 предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 5, принадлежат множеству X^* решений задачи 5 и последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ почти наверное сходится.

Замечание 5. Если существует способ вычисления почти градиента $g^0(x)$ функции f_0 в точке x, то вектор $\theta(x^k, k)$ в (5.39) можно определять по формуле

$$\theta(x^k, k) = g^0(\tilde{x}^k),$$

где $\tilde{x}^k$ — реализация случайной точки, равномерно распределенной в n-мерном кубе со стороной α_k и центром в точке x^k.

6. Стохастический аналог метода возможных направлений

Задача 6. Найти $\arg\min_{x \in X} f_0(x)$ для множества X, которое определяется неравенствами

$$f_i(x) \leqslant b_i, \quad i \in I_1;$$
$$(a^i, x) \leqslant b_i, \quad i \in I_2;$$
$$0 \leqslant x_i \leqslant c_i, \quad i = 1, \ldots, n,$$

где

$$I_1 \subset I \triangleq \{1, \ldots, m\}, \quad I_2 = I - I_1.$$

Предположения 6. (*i*) — функции f_0 и f_i, $i \in I_1$, выпуклые и непрерывно дифференцируемы; (*ii*) — множество X ограничено.

Стохастический аналог метода возможных направлений можно применить в тех случаях, когда либо аналитический вид целевой функции не известен, либо значения данных функций вычисляются с ошибками. Для получения направления движения h^k в k-й итерации решают вспомогательную задачу линейного программирования.

Алгоритм 6

Начало. I. Выбрать: произвольные начальные приближения $x^0 \in X$, $z^0 \in R^n$; начальные значения шаговых множителей ρ_0, γ_0, удовлетворяющие условиям теоремы 6; положительную константу α; θ_i, $i \in I_1$ — некоторые положительные числа.

II. Положить $k = 0$.

Основной цикл. III. Вычислить

$$\varepsilon_k = \alpha\rho_k.$$

IV. Вычислить следующие множества индексов:

$$I_1(x^k, \varepsilon_k) = \{i \in I_1 \mid b_i - \varepsilon_k \leqslant f_i(x^k) \leqslant b_i\};$$
$$I_2(x^k, \varepsilon_k) = \{i \in I_2 \mid b_i - \varepsilon_k \leqslant (a^i, x^k) \leqslant b_i\};$$
$$I_3(x^k, \varepsilon_k) = \{j \in \{1, \ldots, n\} \mid 0 \leqslant x_j^k \leqslant \varepsilon_k\};$$
$$I_4(x^k, \varepsilon_k) = \{j \in \{1, \ldots, n\} \mid c_j - \varepsilon_k \leqslant x_j^k \leqslant c_j\}.$$

V. Вычислить вектор $h = h^k$ — решение вспомогательной задачи линейного программирования:
найти $\max\limits_{(h,\delta)} \delta$ при ограничениях

$$(\nabla f_i(x^k), h) + \theta_i\delta \leqslant 0, \quad i \in I_1(x^k, \varepsilon_k);$$
$$(a^i, h) \leqslant 0, \quad i \in I_2(x^k, \varepsilon_k);$$
$$h_j \geqslant 0, \quad j \in I_3(x^k, \varepsilon_k);$$
$$h_j \leqslant 0, \quad j \in I_4(x^k, \varepsilon_k);$$
$$-1 \leqslant h_j \leqslant 1, \quad j \in \{1, \ldots, n\};$$
$$(z^k, h) + \delta \leqslant 0.$$

VI. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

VII. Вычислить вектор ξ^k — реализацию случайного вектора ξ, условное математическое ожидание которого равно

$$E(\xi/(x^0, z^0), \ldots, (x^k, z^k)) = \beta\nabla f_0(x^k) + r^k,$$

где β — некоторое положительное число; r^k — ограниченный случайный вектор ошибки, измеримый относительно σ-подалгебры, индуцированной величинами $(x^0, z^0), \ldots, (x^k, z^k)$.

VIII. Вычислить вектор

$$z^{k+1} = z^k + \gamma_k(\xi^k - z^k).$$

IX. Вычислить шаговые множители ρ_{k+1} и γ_{k+1}, удовлетворяющие условиям теоремы 6.

X. Положить $k = k + 1$ и перейти к шагу III.

Теорема 6. *Пусть выполнены предположения 6 и, кроме того, имеют место условия: (i) — градиент функции f_0 удовлетворяет условию Липшица*

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \mu\|x - y\|, \quad \mu < \infty;$$

(*ii*) — *для задачи 6 выполнено условие Слейтера;* (*iii*) — *константа* α *такова, что* x^k, $k = 0, 1, \ldots,$ *принадлежит допустимой области* X; (*iv*) — *шаговые множители* ρ_k и γ_k, *измеримые относительно* σ*-подалгебры* $\mathcal{B}_k$, *индуцированной семейством случайных величин* $(x^0, z^0, \ldots, x^k, z^k)$ *таковы, что для некоторых чисел* λ_k

$$\gamma_k \|r^k\| + 2\rho_k - \lambda_k\gamma_k \leqslant 0 \text{ п. н};$$

$$\sum_{k=0}^{\infty} E\rho_k^2 < \infty, \quad \sum_{k=0}^{\infty} E\gamma_k^2 < \infty;$$

$$\sum_{k=0}^{\infty} E(\gamma_k \|r^k\| + \rho_k)\lambda_k < \infty;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad E(\|\xi^k\|^2/\mathcal{B}_k) < \infty, \quad E\|z^0\|^2 < \infty.$$

Тогда последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 6, такова, что* $\{f_0(x^k)\}_{k=0}^{\infty}$ *сходится почти наверное к* $f_0^* \triangleq \min_{x \in X} f_0(x)$.

7. Методы возможных направлений для отыскания точек локальных минимумов невыпуклой функции на невыпуклом множестве

З а д а ч а 7. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X = \{x \mid f_j(x) \geqslant 0, \quad j = 1, \ldots, m, \quad x \in R^n\}.$$

Предположения 7. (*i*) — функции $f_i(x)$, $i = 0, 1, \ldots, m$ — непрерывно дифференцируемы; (*ii*) — $\inf_{x \in X} f_0(x) > -\infty$.

На каждой итерации приводимого здесь алгоритма решается (приближенно) некоторая вспомогательная задача выпуклого программирования (которая может быть заменена задачей линейного программирования). Бесконечная последовательность приближений $\{x^k\}_{k=0}^{\infty}$ сходится к множеству стационарных точек задачи 7, содержащему точки, в которых выполняются необходимые условия локального минимума.

Алгоритм 7

Н а ч а л о. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Выбрать величину ε_0 из полуинтервала (0, 1].

III. Положить $k = 0$.

О с н о в н о й ц и к л. IV. Вычислить множество индексов

$$\mathcal{J}(x^k, \varepsilon_k) = \{j \mid 0 \leqslant f_j(x^k) \leqslant \varepsilon_k, \quad j \in [1:m]\},$$

и положить $\mathcal{J} = \mathcal{J}(x^k, \varepsilon_k)$.

V. Решая вспомогательную задачу,

найти $\max_{(h,\sigma)} \sigma$ **(5.40)**

при ограничениях

$$(\nabla f_j(x^k),\ h)+\sigma\leqslant 0,\quad j\in\mathcal{J};$$
$$-(\nabla f_0(x^k),\ h)+\sigma\leqslant 0; \qquad (5.41)$$
$$(h,\ h)\leqslant 1,$$

вычислить допустимые значения σ_k и h^k, $\| h^k \| = 1$, такие, что

$$\sigma_k \geqslant \xi_k \tilde{\sigma}(x^k,\ \varepsilon_k),$$

где $0<\xi\leqslant\xi_k\leqslant 1$; $\tilde{\sigma}(x^k,\ \varepsilon_k)$, $\tilde{h}(x^k,\ \varepsilon_k)$ — точные решения вспомогательной задачи (5.40) — (5.41).

VI. Если $\sigma_k \geqslant \varepsilon_k$, то перейти к шагу VII, если $\sigma_k = 0$, то перейти к шагу X; иначе (если $0 < \sigma_k < \varepsilon_k$) положить

$$x^{k+1} = x^k;\quad \varepsilon_{k+1} = \gamma_k\varepsilon_k;\quad k = k+1,$$

где $0 < \gamma_k \leqslant \gamma < 1$, и перейти к шагу IV.

VII. Решая задачу одномерной минимизации функции $f_0(x^k - \rho h^k)$ на отрезке $[0,\ \zeta_k]$, вычислить величину ρ_k, удовлетворяющую условиям

$$f_0(x^k-\rho_k h^k)\leqslant (1-\lambda_k) f_0(x^k)+\lambda_k w_k,\quad 0<\lambda\leqslant\lambda_k\leqslant 1;$$
$$w_k=\inf_{0\leqslant\rho\leqslant\zeta_k} f_0(x^k-\rho h^k),$$

где ζ_k — длина наибольшего отрезка $[x^k,\ x^k - \zeta_k h^k]$, принадлежащего множеству X.

VIII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

IX. Положить $\varepsilon_{k+1} = \varepsilon_k$, $k = k + 1$ и перейти к шагу IV.

X. Вычислить множество индексов

$$\mathcal{J}(x^k,\ 0) = \{ j \mid f_j(x^k) = 0,\quad j \in [1:m]\},$$

и положить $\mathcal{J} = \mathcal{J}(x^k,\ 0)$.

XI. Найти величину $\tilde{\sigma}(x^k,\ 0)$ — решение вспомогательной задачи (5.40) — (5.41).

XII. Если $\tilde{\sigma}(x^k,\ 0) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу XIII.

XIII. Положить $x^{k+1} = x^k$; $\varepsilon_{k+1} = \gamma_k\varepsilon_k$, где $0 < \gamma_k \leqslant \gamma < 1$; $k = k + 1$ и перейти к шагу IV.

Теорема 7. *Если выполняются предположения 7 и* (***iii***) *— функции $f_i(x)$, $i = 0, 1, \ldots, m$, принадлежат классу $C^{1,1}(X)$ (если $\psi(x) \in C^{1,1}(X)$, то существует число $\alpha > 0$ такое, что для любого отрезка $[x,\ y]$, целиком принадлежащего множеству X, будет $\| \nabla\psi(x) - \nabla\psi(y) \| \leqslant \alpha \| x - y \|$);* (***iv***) *— существует такое число $\delta > 0$, что $\| \nabla f_j(x) \| \leqslant \delta$ для всех $x \in X$, $j = 1, \ldots, m$;* (***v***) *— множество*

$$X^* = \{x^* \mid \tilde{\sigma}(x^*,\ 0) = 0,\quad x^* \in X\}$$

непусто (здесь $\tilde{\sigma}(x^*, 0)$ *— решение вспомогательной задачи* (5.40) — (5.41) *при* $x^k = x^*$, $\mathcal{I} = \mathcal{I}(x^*, 0)$, *где*

$$\mathcal{I}(x^*, 0) = \{j \mid f_j(x^*) = 0, \quad j \in [1:m]\});$$

(*vi*) *— последовательность* x^k, $k = 0, 1, \ldots$, *порожденная алгоритмом 7, компактна, то в случае бесконечной последовательности* x^k, $k = 0, 1, \ldots$, *справедливо предельное соотношение*

$$\lim_{k \to \infty} \inf_{x^* \in X^*} \|x^k - x^*\| = 0.$$

Замечание 7. Если во вспомогательной задаче (5.40) — (5.41) вместо ограничения $(h, h) \leqslant 1$ поставить ограничение

$$\max_{j \in [1:n]} |h_j| \leqslant 1 \tag{5.42}$$

(таким образом вспомогательная задача становится задачей линейного программирования), то для такого алгоритма теорема 7 остается в силе.

Библиографические указания. Пункт 1 написан на основании работ [285, 563, 172, 574, 175], пункты 2, 3 — на основании работ [320, 285]. При написании пункта 4 использовалась работа [517], пункты 5 и 6 основаны на работах [95, 14], пункт 7 написан на основании работ [37, 8, 188, 189, 167, 572].

В работе [447] изучается обобщенный алгоритм возможных направлений, объединяющий известные алгоритмы Зойтендейка и Э. Полака. В работе [505] разработана общая теория сходимости алгоритмов, использующих ε-параметр, для устранения зигзагоподобного движения в методах возможных направлений.

5.8. Методы центров

Задача 0. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in R^n\}.$$

Предположения 0. (*i*) — функции f_j, $j = 0, 1, \ldots, m$ — непрерывно дифференцируемы; (*ii*) — существует такая точка $x^0 \in X$, что множество

$$X'(x^0) = \{x \mid f_0(x) - f_0(x^0) \leqslant 0, \quad f_j(x) \leqslant 0, \quad j = 1, \ldots, m,\}$$

компактно и имеет внутренность.

Методы центров по своей идее тесно связаны как с методами штрафных функций, так и с методами возможных направлений — их можно трактовать как не содержащие параметров барьерные методы, основанные на внутренних штрафных функциях, или как не содержащие параметров методы возможных направлений.

В методах центров по заданной точке x^k строится следующая точка x^{k+1} в «центре» множества $X'(x^k)$, т. е. в некотором множестве точек, наиболее «удаленных» от границы множества $X'(x^k)$. В качестве функции расстояния между точками x и x' можно взять

функцию

$$d(x', x) = \max\{f_0(x') - f_0(x); \quad f_j(x'), \quad j = 1, \ldots, m\}. \quad (5.43)$$

Другие способы выбора функции расстояния можно найти в [442, 497, 564]. Следующая точка x^{k+1} вычисляется из условия

$$d(x^{k+1}, x^k) = \min_x \{d(x, x^k) \mid x \in R^n\}. \quad (5.44)$$

Таким образом, как и методы штрафных функций, методы центров разбивают решение задачи 0 на последовательность задач минимизации без ограничений.

1. Основной вариант

Алгоритм 1

Начало. I. Найти точку $x^0 \in X$, удовлетворяющую условию *(ii)* предположения 0.

II. Положить $k = 0$.

Основной цикл. III. Найти такую точку $x^1 \in R^n$, что

$$d(x^1, x^k) = \min_x d(x, x^k). \quad (5.45)$$

IV. Если $d(x^1, x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу V.

V. Положить $x^{k+1} = x^1$, $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполнены предположения 0 и пусть, кроме того, (iii) — множество X имеет внутренность и ее замыкание совпадает с X, т. е.* $\overline{\text{int}\, X} = X$; *(iv)* — $f_j(x) < 0$ *для всех* $x \in \text{int}\, X$, $j = \overline{1, m}$. *Если построенная алгоритмом 1 последовательность* $\{x^k\}_{k=0}^{\infty}$ *бесконечна, то она имеет предельные точки и все они оптимальны для задачи 0, а если последовательность* $\{x^k\}$ *конечна, то ее последний элемент также оптимален для задачи 0.*

На практике было обнаружено, что алгоритм 1 — неэффективный в основном из-за трудностей вычисления точки x^1, определяемой согласно (5.45), которые требуют очень больших затрат времени. Это привело к необходимости разработки методов, которые прерывали бы отыскание точки x^1, когда уже найдена точка x'', являющаяся «ε-центром», где ε выбирается по возможности малым. Опишем модифицированный метод, прерывающий поиск центра x^1 множества $X'(x^k)$ после единственной итерации процедуры поиска.

2. Модифицированный метод центров

Алгоритм 2

Начало. I. Найти точку $x^0 \in X$, для которой выполнено условие *(ii)* предположения 0.

II. Положить $k = 0$.

О с н о в н о й   ц и к л. III. Положить $x = x^k$.

IV. Вычислить решение $h_0 = h_0(x)$, $h = h(x)$ задачи линейного программирования:
найти

$$\min_{h_0, h} h_0 \tag{5.46}$$

при ограничениях

$$-h_0 + (\nabla f_0(x),\ h) \leqslant 0; \tag{5.47}$$

$$-h_0 + f_j(x) + (\nabla f_j(x),\ h) \leqslant 0, \quad j = 1, \ldots, m; \tag{5.48}$$

$$|h_i| \leqslant 1, \quad i = 1, \ldots, n, \quad (h = (h_1, \ldots, h_n)). \tag{5.49}$$

V. Если $h_0(x) = 0$, то положить $x^* = x^k$ и прекратить вычисления (если x^* — оптимальное решение задачи 0, то

$$h_0(x^*) \triangleq \min_{h \in S} (\max \{(f_0(x^*),\ h);\ f_j(x^*) + \\ + (\nabla f_j(x^*),\ h), \quad j = 1, \ldots, m\}) = 0,$$

где

$$S \triangleq \{h \in R^n \mid |h_i| \leqslant 1, \quad i = 1, \ldots, n\}).$$

иначе перейти к шагу VI.

VI. Вычислить наименьшее положительное значение $\mu(x)$, удовлетворяющее условию

$$d(x + \mu(x) h(x),\ x) = \min_{\mu \geqslant 0} d(x + \mu h(x),\ x), \tag{5.50}$$

где $d(\cdot, \cdot)$ определяется формулой (5.43).

VII. Положить $x^{k+1} = x + \mu(x) h(x)$, $k = k + 1$ и перейти к шагу III.

Замечание 2. Вектор $h(x)$, вычисляемый на шаге IV алгоритма 2, неоднозначно определяется из решения задачи (5.46) — (5.49), поскольку решение задачи линейного программирования может не быть единственным. Считается, что $h(x)$ — это любой вектор, являющийся оптимальным решением задачи (5.46) — (5.49); его можно получить методами, описанными в гл. 4.

Замечание 2'. Точку $x^0 \in X$, т. е. начальное приближение на шаге I алгоритма 2 можно вычислить, применяя алгоритм 2 к следующей задаче (в пространстве векторов (x_0, x)):

$$\min_{(x_0, x)} \{x_0 \mid f_j(x) - x_0 \leqslant 0, \quad j = 1, \ldots, m\},$$

причем в качестве допустимого начального решения для этой задачи можно взять $\bar{x}^0 \triangleq (x_0^0, x^0) \in R^{n+1}$, где точка $x^0 \in R^n$ — произвольная и $x_0^0 = \max\{f_j(x^0),\ j \in \{1, \ldots, m\}\}$. Найдется такое конечное целое число i, что $\bar{x}^i = (x_0^i, x^i)$ будет удовлетворять условиям $x_0^i < 0$, $x^i \in X$. Тогда положить $x^0 = x^i$ и перейти к решению задачи 0 с помощью алгоритма 2

Теорема 2. Пусть выполнены предположения 0. Тогда последовательность x^0, x^1, ..., построенная алгоритмом 2, либо конечна и ее последний элемент $x^* = x^k$ удовлетворяет условию $h_0(x^*) = 0$, либо бесконечна и каждая ее предельная точка x^* удовлетворяет условию $h_0(x^*) = 0$.

3. Реализация модифицированного метода центров с использованием метода золотого сечения

Рассмотрим, как заменить принципиальную операцию (5.50) в алгоритме 2 другой, которую можно было бы реализовать на цифровой ЭВМ. Простейший — это случай, когда все функции $f_j(\cdot)$, $j = 0, 1, \ldots, m$ — выпуклы. Легко видеть, что в этом случае функция $d(\cdot, x)$, определяемая согласно (5.43) при $x \in X$, также выпуклая.

Алгоритм 3

Начало. I. Выбрать числа $\bar{\varepsilon}_0 > 0$, $\rho \geqslant 1$ и $\eta > 1$ (обычно $\eta = 2$).

II. Вычислить начальное приближение $x^0 \in X$.

III. Положить $k = 0$.

Основной цикл. IV. Положить $x = x^k$.

V. Решить задачу линейного программирования (5.46) — (5.49) и получить $(n+1)$-мерный вектор $(h_0(x), h(x))$.

VI. Если $h_0(x) = 0$, то положить $x^{k+1} = x^k$ и прекратить вычисления; иначе перейти к шагу VII.

Примечание. Следующие далее шаги VII—X заменяют шаг VI алгоритма 2 при вычислении величины $\mu(x)$.

VII. Для $\lambda \geqslant 0$ положить

$$\theta(\lambda) = d(x + \lambda h(x), x). \tag{5.51}$$

VIII. Положить $\varepsilon_i = \bar{\varepsilon}_0$ и $i = 0$.

IX. Использовать для вычисления $\bar{\mu}$ алгоритм поиска минимума одномерной функции по методу золотого сечения (см. § 2.4, алгоритм 2Б), где $\theta(\cdot)$ определяется равенством (5.51), ρ определяется на шаге I и $\varepsilon = \varepsilon_i$.

X. Если $d(x + \bar{\mu}h(x), x) \leqslant -\varepsilon_i$, то положить $\mu(x) = \bar{\mu}$ и перейти к шагу XI; иначе положить $\varepsilon_{i+1} = \varepsilon_i/\eta$, $i = i + 1$ и перейти к шагу IX.

XI. Положить $x^{k+1} = x + \mu(x)h(x)$, $k = k + 1$ и перейти к шагу IV.

Теорема 3. Пусть выполнены предположения 0 и пусть функции $f_j(\cdot)$, $j = 0, 1, \ldots, m$ — выпуклы. Тогда последовательность x^0, x^1, ..., построенная алгоритмом 3, либо конечна и ее последний элемент $x^{k+1} = x^k$ удовлетворяет условию $h_0(x^k) = 0$, либо бесконечна и тогда каждая ее предельная точка x^* удовлетворяет условию $h_0(x^*) = 0$.

Замечание 3. Можно построить другие реализации алгоритма 2,

заменяя операцию (5.50) на шаге VI любым алгоритмом поиска минимума (или стационарной точки) действительной функции одной переменной.

4. Реализация модифицированного метода центров с использованием функций прерывания

На практике алгоритмы изложенного ниже типа эффективны в ситуации, когда они по существу решают много раз одну и ту же задачу. В этом случае следует предварительно потратить определенное время на тщательный подбор «функции прерывания», которая будет использоваться, и попытаться получить алгоритм, действующий быстрее, чем алгоритм 3 с ε-процедурой.

Определение 4. Пусть $N = \{0, 1, 2, \ldots\}$. Будем называть функцию $t : N \to N$ функцией прерывания, если для каждого $i \in N$ найдется такое число $i' \in N$, что $t(k) > i$ при всех $k > i'$.

Алгоритм 4

Начало. I. Вычислить начальное приближение $x^0 \in X$ и положить $k = 0$.

II. Выбрать функции прерывания $t_1(\cdot)$, $t_2(\cdot)$.

Основной цикл. III. Положить $x = x^k$, $i = t_1(k)$.

IV. Решить задачу линейного программирования (5.46) — (5.49) и получить $n+1$-мерный вектор $(h_0(x), h(x))$.

V. Если $h_0(x) = 0$, то положить $x^{k+1} = x^k$ и прекратить вычисления; иначе перейти к шагу VI.

VI. Положить $\theta(\mu) = d(x + \mu h(x), x)$ и использовать для вычисления $\bar{\mu}$ алгоритм 4' — модификацию алгоритма поиска минимума одномерной функции по методу золотого сечения.

VII. Если $f_0(x + \bar{\mu} h(x)) < f_0(x)$, то положить $x^{k+1} = x + \bar{\mu} h(x)$, $k = k+1$ и перейти к шагу III; иначе положить $x^{k+1} = x^k$, $i = t_2(k)$, $k = k+1$ и перейти к шагу VI.

***Алгоритм 4'* (модификация алгоритма поиска по методу золотого сечения)**

I. Выбрать $\rho > 0$, вычислить дроби Фибоначчи

$$F_1 = (3 - \sqrt{5})/2, \quad F_2 = (\sqrt{5} - 1)/2.$$

II. Вычислить $\theta(0)$, $\theta(\rho)$.

III. Если $\theta(\rho) \geqslant \theta(0)$, то положить $a_0 = 0$, $b_0 = \rho$ и перейти к шагу VIII; иначе перейти к шагу IV.

IV. Положить $l = 0$ и $\mu_0 = 0$.

V. Положить $\mu_{l+1} = \mu_l + \rho$.

VI. Вычислить $\theta(\mu_{l+1})$.

VII. Если $\theta(\mu_{l+1}) \geqslant \theta(\mu_l)$, то положить $a_0 = \mu_{l-1}$, $b_0 = \mu_{l+1}$ и перейти к шагу VIII; иначе положить $l = l + 1$ и перейти к шагу V.

Примечание. Теперь минимизирующая точка лежит в $[a_0, b_0]$.

VIII. Положить $j = 0$.

IX. Положить $v_j = a_j + F_1 (b_j - a_j)$, $w_j = a_j + F_2 (b_j - a_j)$.

X. Если $\theta (v_j) < \theta (w_j)$, то положить $a_{j+1} = a_j$, $b_{j+1} = w_j$ и перейти к шагу XI; иначе положить $a_{j+1} = v_j$, $b_{j+1} = b_j$ и перейти к шагу XI.

XI. Если $j < i$, то положить $j = j + 1$ и перейти к шагу IX; иначе положить $\bar{\mu} = (a_j + b_j)/2$ и прекратить вычисления.

Теорема 4. *Пусть выполнены предположения 0 и пусть функции $f_j(\cdot)$, $j = 0, 1, \ldots, m$ — выпуклы. Тогда последовательность $x^0, x^1, \ldots$ $\ldots$, построенная алгоритмом 4, либо конечна и ее последний элемент $x^{k+1} = x^k$ удовлетворяет условию $h_0 (x^k) = 0$, либо бесконечна и тогда каждая ее предельная точка x^* удовлетворяет условию $h_0 (x^*) = 0$.*

Библиографические указания. Пункты 1, 2 написаны на основании работ [285, 496, 497], пункты 3, 4 — на основании работы [285].

5.9. Методы чебышевских центров

З а д а ч а 0. Найти $\arg\max\limits_{x \in X} f_0 (x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного выпуклого множества

$$X = \{x \mid (a^i, x) \geqslant \alpha_i, \quad i = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}.$$

Предположение 0. Функция f_0 — вогнутая.

1. Метод чебышевских центров

В методе чебышевских центров каждое последующее приближение x^{k+1} находят решением специальной задачи линейного программирования.

Алгоритм 1

Н а ч а л о. I. Выбрать произвольное начальное приближение $x^1 \in X$.

II. Положить $k = 1$.

О с н о в н о й ц и к л. III. Вычислить обобщенный градиент $\hat{\nabla} f_0 (x^k)$ и положить

$$a^{m+k} = \hat{\nabla} f_0 (x^k), \quad \alpha_{m+1} = (\hat{\nabla} f_0 (x^k), x^k).$$

IV. Вычислить точку $x^{k+1} \in X$, удовлетворяющую условию

$$\max_{x \in X} \min_{m+1 \leqslant i \leqslant m+k} ((a^i, x) - \alpha_i) = \min_{m+1 \leqslant i \leqslant m+k} ((a^i, x^{k+1}) - \alpha_i).$$

V. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Если выполняется предположение 0 и функция $\| \hat{\nabla} f_0 (x) \|$ ограничена на X, то последовательность $\{x^k\}_{k=1}^{\infty}$, порожденная алгоритмом 1, такова, что каждая предельная точка ее монотонной подпоследовательности $\{x^{l_k}\}_{k=1}^{\infty}$ относительно функции $f_0 (x)$ является решением задачи 0.*

2. Модифицированный метод чебышевских центров

В модифицированном методе чебышевских центров на каждой итерации производится выбрасывание лишних ограничений, при этом на каждой итерации требуется решать задачу максимизации суммы кусочно линейной и квадратичной функций.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^1 \in X$.

II. Положить $Q_1 = \{m+1\}$.

III. Положить $k = 1$.

Основной цикл. IV. Вычислить обобщенный градиент $\hat{\nabla} f_0(x^k)$ и положить

$$a^{m+k} = \hat{\nabla} f_0(x^k), \quad \alpha_{m+k} = (\hat{\nabla} f_0(x^k),\ x^k).$$

V. Вычислить точку $x^{k+1} \in X$, удовлетворяющую условию

$$\max_{x \in X} \min_{i \in Q_k} \left((a^i,\ x) - \alpha_i - \frac{1}{2}\frac{1}{\sqrt{k}}(x,\ x)\right) =$$
$$= \min_{i \in Q_k} \left((a^i,\ x^{k+1}) - \alpha_i - \frac{1}{2}\frac{1}{\sqrt{k}}(x^{k+1},\ x^{k+1})\right).$$

VI. Положить

$$\nu_k = \min_{i \in Q_k} \left((a^i,\ x^{k+1}) - \alpha_i - \frac{1}{2}\frac{1}{\sqrt{k}}(x^{k+1},\ x^{k+1})\right).$$

VII. Найти минимальное подмножество G_k множества Q_k, т. е. такое, что

$$\max_{x \in X} \min_{i \in G_k} \left((a^i,\ x) - \alpha_i - \frac{1}{2}\frac{1}{\sqrt{k}}(x,\ x)\right) = \nu_k;$$

$$\max_{x \in X} \min_{i \in G_k \setminus \{j\}} \left((a^i,\ x) - \alpha_i - \frac{1}{2}\frac{1}{\sqrt{k}}(x,\ x)\right) > \nu_k, \quad \forall j \in G_k.$$

VIII. Положить $Q_{k+1} = Q_k \cup \{m+k+1\}$.

IX. Положить $k = k+1$ и перейти к шагу IV.

Теорема 2. *Если выполняется предположение 0 и функция $\|\hat{\nabla} f_0(x)\|$ ограничена на X, то последовательность $\{x^k\}_{k=1}^{\infty}$, порожденная алгоритмом 2, такова, что ее монотонная подпоследовательность $\{x^{i_k}\}_{k=1}^{\infty}$ относительно функции $f_0(x)$ бесконечна и каждая предельная точка x^* является решением задачи 0. Кроме того,*

$$0 \leqslant f_0(x^*) - f_0(x^{i_k}) = O(1/\sqrt{i_k}).$$

Библиографические указания. При написании параграфа использовались работы [308, 309]. Дополнительные сведения о методах чебышевских центров можно найти в работах [307, 46].

5.10. Методы типа Ньютона

1. Метод типа Ньютона с регулировкой шага

Задача 1. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0: R^n \to R^1$ и заданного выпуклого компактного множества $X \subset R^n$.

Предположение 1. Функция f_0 — дважды непрерывно дифференцируема на выпуклом компактном множестве X.

Алгоритм 1

Начало. I. Задать начальное приближение $x^0 \in X$.

II. Выбрать константы $\varepsilon \in (0, 1)$, $\beta \in (0, 1)$ (рекомендуется $\varepsilon = 1/2$, $\beta = 0{,}8$).

III. Положить $k = 0$.

Основной цикл. IV. Положить $x = x^k$.

V. Вычислить матрицу вторых производных $\nabla^2_{xx} f_0(x)$ и градиент $\nabla f_0(x)$.

VI. Вычислить точку y^k, обеспечивающую минимум квадратичной функции $\psi_k(y)$ на множестве X, где

$$\psi_k(y) \triangleq (\nabla f_0(x),\ y - x) + \frac{1}{2}(\nabla^2_{xx} f_0(x)(y - x),\ y - x). \quad (5.52)$$

Если $\psi_k(y^k) = 0$, то прекратить вычисления (в этом случае y^k — решение задачи 1); иначе перейти к шагу VII.

VII. Положить $h^k = y^k - x$.

VIII. Положить $\rho_k = 1$.

IX. Если выполняется неравенство

$$f_0(x + \rho_k h^k) - f_0(x) \leqslant \varepsilon \rho_k \psi_k(y^k), \quad (5.53)$$

то перейти к шагу XI; иначе перейти к шагу X.

X. Положить $\rho_k = \beta\rho_k$ и перейти к шагу IX.

XI. Положить $x^{k+1} = x + \rho_k h^k, k = k + 1$ и перейти к шагу IV.

Теорема 1. *Если минимизируемая функция f_0 выпукла и дважды непрерывно дифференцируема на выпуклом компактном множестве X, то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, выполняются условия:*

а) $\{f_0(x^k)\}_{k=0}^{\infty}$ *монотонно убывает;*

б) $\lim_{k\to\infty} f_0(x^k) = \min_{x\in X} f_0(x)$.

Если, кроме того,

$$\alpha_1 \|y\|^2 \leqslant (\nabla^2_{xx} f_0(x) y,\ y) \leqslant \alpha_2 \|y\|^2,\ \alpha_1 > 0,\ x \in X,\ y \in R^n, \quad (5.54)$$

то последовательность $\{x^k\}_{k=0}^{\infty}$ сходится к решению задачи 1 со сверхлинейной скоростью, т. е.

$$\|x^{l+i} - x^*\| \leqslant \nu_1 \lambda_l \ldots \lambda_{l+i}, \quad (5.55)$$

где $l, \nu_1 < \infty$, $\lambda_{l+i} < 1$ при любом $i \geqslant 0$, $\lambda_j \to 0$ при $j \to \infty$.

Теорема 1'. Если выполняется условие (5.54) и матрица $\nabla^2_{xx} f_0(x)$ на выпуклом компактном множестве X удовлетворяет условию Липшица с константой γ, т. е.

$$\| \nabla^2_{xx} f_0(x) - \nabla^2_{xx} f_0(y) \| \leqslant \gamma \| x - y \|, \quad \forall x,\ y \in X, \quad \gamma < \infty,$$

то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к решению x^* задачи 1 с квадратичной скоростью, т. е.

$$\| x^{l+i} - x^* \| \leqslant \nu_2 \beta_l^{2^i}, \quad \nu_2 < \infty, \quad l < \infty, \quad \beta_l < 1.$$

Приводимый ниже алгоритм является принципиальной схемой для алгоритма 1.

Алгоритм 1'

I—VII. Шаги I—VII такие, как и в алгоритме 1.

VIII. Вычислить число $\rho_k \in [0, 1]$, удовлетворяющее условию

$$f_0(x + \rho_k h^k) = \min_{\rho \in [0,1]} f_0(x + \rho h^k).$$

IX. Положить $x^{k+1} = x + \rho_k h^k$, $k = k + 1$ и перейти к шагу IV.

Теорема 1''. *Если выполняется условие (5.54) и множество X — выпуклый компакт, то бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1', сходится к решению задачи 1 со сверхлинейной скоростью, т. е. имеет место оценка (5.55), причем $\rho_k \to 1$ при $k \to \infty$.*

2. Метод типа Ньютона при наличии возмущений

З а д а ч а 2. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества X.

Предположения 2. (i) — множество X — замкнутое и выпуклое; (ii) — функция $f_0(x)$ дважды непрерывно дифференцируема на множестве X.

На k-й итерации алгоритма 2 функция $f_0(x) - f_0(x^k)$ в точке x^k аппроксимируется квадратичным функционалом

$$f^k(x) = \frac{1}{2} (A_k (x - x^k),\ x - x^k) + (h^k,\ x - x^k), \tag{5.56}$$

где A_k — приближение для $\nabla^2_{xx} f_0(x^k)$; h^k — приближение для $\nabla f_0(x^k)$; в качестве x^{k+1} выбирается точка, для которой приближенно выполняется условие минимума $f^k(x)$ на множестве X

$$\left(\nabla f^k(x^{k+1}),\ \frac{x^{k+1} - x}{\| x^{k+1} - x \|} \right) \triangleq \left(A_k (x^{k+1} - x^k) + h^k,\ \frac{x^{k+1} - x}{\| x^{k+1} - x \|} \right) \leqslant \sigma_k$$

для всех $x \in X$.

Алгоритм 2

Н а ч а л о. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II Положить $k = 0$.

Основной цикл. III. Вычислить матрицу A_k — приближение для $\nabla^2_{xx} f_0(x^k)$.

IV. Вычислить вектор h^k — приближение для $\nabla f_0(x^k)$.

V. Вычислить точку $x^{k+1} \in X$, для которой выполняется неравенство

$$\left(A_k(x^{k+1} - x^k) + h^k, \frac{x^{k+1} - x}{\| x^{k+1} - x \|}\right) \leqslant \sigma_k, \quad \sigma_k \geqslant 0$$

для всех $x \in X$. Вычисление точки x^{k+1} сводится к нахождению приближенного минимума квадратичного функционала (5.56) на множестве X.

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполняются предположения 2 и (iii) — матрица вторых производных $\nabla^2_{xx} f_0(x)$ удовлетворяет условию Липшица с константой γ в области*

$$\overline{X}_0 = \{x \mid \| x - x^0 \| \leqslant \rho, \quad x \in X\};$$

(*iv*) —

$$(\nabla^2_{xx} f_0(x) y, y) \geqslant m_1 \| y \|^2, \quad m_1 > 0.$$

Тогда: 1) *если выполняются условия*

(*v*) —

$$\| A_k - \nabla^2_{xx} f_0(x^k) \| \leqslant c_1;$$

(*vi*) —

$$\| h^k - \nabla f_0(x^k) \| \leqslant c_2 \| x^k - x^{k+1} \|;$$

(*vii*) —

$$\sigma_k \leqslant c_3 \| x^k - x^{k+1} \|, \quad m_1 > c_4, \quad c_4 = c_1 + c_2 + c_3;$$

(*viii*) —

$$q = \beta + \gamma c_4^2 \| x^1 - x^0 \| / 2(m_1 - c_4) < 1, \quad \beta = c_4/(m_1 - c_4);$$

(*ix*) —

$$\rho \geqslant r_1 \triangleq \| x^1 - x^0 \| / (1 - q),$$

то существует точка минимума $x^* \in \overline{X}_0$ и последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, такова, что

$$\| x^k - x^* \| \leqslant r_1 q^k, \quad \| x^k - x^* \| = O(\beta^k);$$

2) если выполняются условия

(*x*) —

$$\| A_k - \nabla^2_{xx} f_0(x^k) \| \leqslant c_1 \| x^k - x^{k+1} \|;$$

(*xi*) —

$$\| h^k - \nabla f_0(x^k) \| \leqslant c_2 \| x^k - x^{k+1} \|^2;$$

(*xii*) —

$$\sigma_k \leqslant c_3 \| x^k - x^{k+1} \|^2, \ m_1 > c_4 \| x^1 - x^0 \|, \quad c_4 = c_1 + c_2 + c_3;$$

(*xiii*) —

$$q \triangleq (c_4 + \gamma/2) \| x^1 - x^0 \| / (m_1 - c_4 \| x^1 - x^0 \|) < 1;$$

(*xiv*) —

$$\rho \geqslant (\| x^1 - x^0 \| / q) \psi_0(q),$$

то существует точка минимума $x^* \in \overline{X}_0$ и последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, такова, что

$$\|x^k - x^*\| \leqslant (\|x^1 - x^0\|/q)\,\psi_k(q),$$

где

$$\psi_k(q) = \sum_{i=k}^{\infty} q^{2^i}, \quad k = 0, 1, \ldots .$$

3. Квазиньютоновские методы

Задача 3. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X \triangleq \{x \mid f_i(x) \leqslant 0, \quad i = 1, \ldots, m, \quad x \in R^n\},$$

где $f_i : R^n \to R^1$, $i = 1, \ldots, m$ — заданные функции.

Предположения 3. Функции $f_j(x)$, $j = 0, 1, \ldots, m$ — дважды дифференцируемые в R^n.

На k-й итерации алгоритма требуется вычислять матрицу, близкую к матрице вторых производных по x, функции Лагранжа задачи 3 (в частности, можно использовать конечно-разностную аппроксимацию гессиана по x функции Лагранжа). Следующее $(k+1)$-е приближение (x^{k+1}, u^{k+1}) по основным и двойственным переменным задачи 3 находится как решение некоторой вспомогательной задачи квадратичного программирования.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $(x^0, u^0) \in R^n \times R^m$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить матрицу $G(x^k, u^k)$, удовлетворяющую условиям теоремы 3.

IV. Найти точку Куна — Таккера (x^{k+1}, u^{k+1}), решая следующую задачу квадратичного программирования:

$$\text{найти } \arg\min_{x} \left[(\nabla f_0(x^k),\ x - x^k) + \frac{1}{2}(x - x^k,\ G(x^k, u^k)(x - x^k))\right] \tag{5.57}$$

при ограничениях

$$f_i(x^k) + (\nabla f_i(x^k),\ x - x^k) \leqslant 0, \quad i = 1, \ldots, m.$$

Если решение задачи (5.57) не единственное, то в качестве (x^{k+1}, u^{k+1}) выбрать точку, ближайшую к (x^k, u^k).

V. Положить $k = k + 1$ и перейти к шагу III.

Теорема 3. *Пусть (i) — $(\overline{x}, \overline{u})$ — точка Куна — Таккера, удовлетворяющая достаточным условиям второго порядка для задачи 3 (точка $(\overline{x}, \overline{u})$ из $R^n \times R^m$ удовлетворяет достаточным условиям*

второго порядка, если она удовлетворяет условиям Куна — Таккера первого порядка и если $\nabla^2_{xx}\varphi(\bar{x}, \bar{u})\, y > 0$ *для каждого ненулевого* $y \in \boldsymbol{R}^n$, *удовлетворяющего условиям*

$$\nabla f_j(\bar{x})\, y = 0, \quad j \in \{j \mid \bar{u}_j > 0, \quad j = 1, \ldots, m\};$$

$$\nabla f_j(\bar{x})\, y \leqslant 0, \quad j \in \{j \mid \bar{u}_j = 0, \quad f_j(\bar{x}) = 0, \quad j \in [1:m]\},$$

где $\nabla^2_{xx}\varphi(\bar{x}, \bar{u})$ — *матрица вторых производных по x функции Лагранжа*

$$\varphi(x, u) = f_0(x) + \sum_{i=1}^{m} u_i f_i(x), \quad u = (u_1, \ldots, u_m),$$

вычисленная в точке $(\bar{x}, \bar{u})$),

а также условию строгой дополняющей нежесткости и условию линейной независимости градиентов активных ограничений (точка $(\bar{x}, \bar{u})$ *удовлетворяет условию строгой дополняющей нежесткости, если* $\bar{u}_j > 0$ *или* $f_j(\bar{x}) < 0$ *для* $j = 1, \ldots, m$); (*ii*) — $f_j(x)$, $j = 0, 1, \ldots, m$, *имеют вторые производные, непрерывные по Липшицу в открытой окрестности точки* $\bar{x}$; (*iii*) — *матрицы* $G(x^k, u^k)$ *на шаге III алгоритма 3 удовлетворяют одному из условий:*

а)

$$\|G(x^k, u^k) - \nabla^2_{xx}\varphi(x^k, u^k)\| \leqslant \frac{1}{10\beta},$$

где

$$\beta = \frac{3}{2}\|(\nabla_z h(z^k))^{-1}\|;$$

$$h(z) = \begin{pmatrix} \nabla f_0(x) + \nabla f(x)\, u \\ u_1 f_1(x) \\ \vdots \\ u_m f_m(x) \end{pmatrix}, \quad \begin{aligned} & f(x) = (f_1(x), \ldots, f_m(x)), \\ & z = (x, u), \\ & z^k = (x^k, u^k); \end{aligned}$$

б)

$$\|G(x^k, u^k) - \nabla^2_{xx}\varphi(x^k, u^k)\| \leqslant \frac{1}{10\beta},$$

$$\left\| \frac{(G(x^k, u^k) - \nabla^2_{xx}\varphi(x^k, u^k))(x^{k+1} - x^k)}{\|z^{k+1} - z^k\|} \right\| \to 0 \text{ при } k \to \infty \tag{5.58}$$

в)

$$G(x^k, u^k) = \nabla^2_{xx}\varphi(x^k, u^k). \tag{5.59}$$

Тогда существуют положительные числа δ_1 *и* δ_2 *такие, что если начальное приближение* (x^0, u^0) *выбрать из условий*

$$\|(x^0, u^0) - (\bar{x}, \bar{u})\| < \delta_1;$$

$$\|G(x^0, u^0) - \nabla^2_{xx}\varphi(x^0, u^0)\| < \delta_2,$$

то последовательность $\{(x^k, u^k)\}_{k=0}^{\infty}$ *существует и сходится к точке* $(\bar{x}, \bar{u})$:

1) *с линейной скоростью сходимости, т. е. существуют числа* $q \in (0, 1)$, $\alpha > 0$, $\bar{k} \geqslant 0$ *такие, что*

$$\|(x^k, u^k) - (\bar{x}, \bar{u})\| \leqslant \alpha q^k \text{ для всех } k \geqslant \bar{k};$$

2) *со сверхлинейной скоростью сходимости, т. е. для каждого сколь угодного малого* $q \in (0, 1)$ *существуют* $\alpha > 0$ *и* $\bar{k} \geqslant 0$ *такие, что*

$$\|(x^k, u^k) - (\bar{x}, \bar{u})\| \leqslant \alpha q^k \text{ для всех } k \geqslant \bar{k},$$

при условии, что матрицы $G(x^k, u^k)$, $k = 0, 1, \ldots,$ *удовлетворяют* (5.58);

3) *с квадратичной скоростью сходимости, т. е. существуют числа* $q \in (0, 1)$, $\alpha > 0$, $\bar{k} \geqslant 0$ *такие, что*

$$\|(x^k, u^k) - (\bar{x}, \bar{u})\| \leqslant \alpha q^{2^k} \text{ для всех } k \geqslant \bar{k}$$

при условии, что матрицы $G(x^k, u^k)$, $k = 0, 1, \ldots,$ *удовлетворяют* (5.59).

Библиографические указания. Пункт 1 написан на основании работ [105, 106, 320, 222], пункт 2 — на основании работы [295], а пункт 3 — на основании работы [476].

Метод Ньютона изучался также в работах [440, 479, 16, 485, 556, 426].

5.11. Методы линеаризации

Методы линеаризации применяются для отыскания решения x^* задачи минимизации нелинейной функции $f_0(x)$ при довольно общих ограничениях в виде равенств $f_j(x) = 0$, $j = 1, \ldots, m$, и неравенств $f_j(x) \leqslant 0$, $j = m + 1, \ldots, m + l$. Решение x^* определяется как предел строящейся последовательности $x^0, x^1, \ldots, x^k, x^{k+1}, \ldots$. Улучшенное $(k + 1)$-е приближение x^{k+1} к решению x^* определяется с помощью решения значительно более простой вспомогательной задачи минимизации функции

$$\hat{f}_0(x) \triangleq (\nabla f_0(x^k), x) + \frac{1}{2}\|x - x^k\|^2 \tag{5.60}$$

при линейных ограничениях

$$\hat{f}_j(x) \triangleq f_j(x^k) + (\nabla f_j(x^k), x - x^k) \leqslant \varepsilon_k, \quad j \in \mathcal{I}_{\delta_k}. \tag{5.61}$$

Вспомогательным множеством $\mathcal{I}_{\delta_k}$ выделяют только те из исходных ограничений, которые желательно удовлетворить на $(k + 1)$-м приближении x^{k+1} (например, те из ограничений, которые «наиболее нарушаются» на k-м приближении x^k). Квадратичная добавка $\frac{1}{2}\|x - x^k\|^2$ гарантирует существование решения $z(\varepsilon, \delta_k)$

вспомогательной задачи на непустом допустимом множестве. Улучшенное $(k+1)$-е приближение x^{k+1} вычисляют по формуле

$$x^{k+1} = x^k + \rho_k (z(\varepsilon, \delta_k) - x^k).$$

Различные способы определения шаговых множителей ρ_k и констант ε, δ_k определяют разные варианты методов линеаризации.

1. Ограничения типа неравенств

Задача 1. Найти $\arg\min\limits_{x \in X} f_0(x)$ для заданной функции $f_0: R^n \to R^1$ и множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j \in \mathcal{J}, \quad x \in R^n\},$$

определенного заданными функциями $f_j: R^n \to R^1$, $j \in \mathcal{J}$.

Предположения 1. (*i*) — функции f_j, $j \in \{0\} \cup \mathcal{J}$ — непрерывно дифференцируемы; (*ii*) — градиенты функций f_j, $j \in \{0\} \cup \mathcal{J}$ удовлетворяют условию Липшица

$$\|\nabla f_j(x) - \nabla f_j(y)\| \leqslant \gamma \|x - y\|, \quad j \in \{0\} \cup \mathcal{J}, \quad \gamma < \infty.$$

Алгоритм 1

Начало. I. Выбрать произвольное приближение $x^0 \in R^n$.

II. Выбрать достаточно большую константу $\alpha > 0$ и константу $\delta > 0$, удовлетворяющие условиям теоремы 1.

III. Выбрать константу $0 < \varepsilon < 1$.

IV. Положить $k = 0$.

Основной цикл. V. Положить $x = x^k$.

VI. Найти множество индексов

$$\mathcal{J}_\delta(x) = \{j \mid f_j(x) \geqslant \max_{j \in \mathcal{J}} f_j(x) - \delta, \quad j \in \mathcal{J}\}.$$

VII. Найти решение $h = h(x)$ задачи квадратичного программирования:

$$\text{найти } \arg\min_h \left((\nabla f_0(x), h) + \frac{1}{2}\|h\|^2\right) \text{ при ограничениях}$$

$$(\nabla f_j(x), h) + f_j(x) \leqslant 0, \quad j \in \mathcal{J}_\delta(x).$$

VIII. Если $h(x) = 0$, то положить $x^* = x$ и прекратить вычисления; иначе перейти к шагу IX.

IX. Положить $i = 0$.

X. Положить $\rho_k = (1/2)^i$.

XI. Если выполняется неравенство

$$f_0(x + \rho_k h(x)) + \alpha \max_{j \in \mathcal{J}} \{0, f_j(x + \rho_k h(x))\} \leqslant$$

$$\leqslant f_0(x) + \alpha \max_{j \in \mathcal{J}} \{0, f_j(x)\} - \rho_k \varepsilon \|h(x)\|^2,$$

то перейти к шагу XII; иначе положить $\bar{i} = \bar{i} + 1$ и перейти к шагу X.

XII. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h(x).$$

XIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если выполнены предположения 1 и существуют константы* $\alpha > 0$ и $\delta > 0$ *такие, что* (*iv*) — *множество*

$$X_\alpha = \{x \mid f_0(x) + \alpha \max_{j \in \mathcal{J}} f_j(x) \leqslant f_0(x^0) + \alpha \max_{j \in \mathcal{J}} f_j(x^0), \quad x \in \boldsymbol{R}^n\},$$

ограничено;

(*v*) — *задача квадратичного программирования — найти* $\arg\min_h \left((\nabla f_0(x),\ h) + \frac{1}{2}(h,\ h)\right)$ *при ограничениях*

$$(\nabla f_j(x),\ h) + f_j(x) \leqslant 0, \quad j \in \mathcal{J}_\delta(x),$$

разрешима относительно $h \in \boldsymbol{R}^n$ *при любом* $x \in X_\alpha$ *и существуют такие множители Лагранжа* $\lambda_j(x)$, $j \in \mathcal{J}_\delta(x)$, *что* $\sum_{j \in \mathcal{J}_\delta(x)} \lambda_j(x) \leqslant \alpha$; (*vi*) — *существует такой индекс* $j \in \mathcal{J}$, *что функция* $f_j(x) \equiv 0$, *то бесконечная последовательность* $\{x^k\}_{k=0}^\infty$, *порожденная алгоритмом 1, обладает следующими свойствами:*

$$\max_{j \in \mathcal{J}} f_j(x^k) \to 0 \ \text{при}\ k \to \infty;$$

любая предельная точка x^* *последовательности* $\{x^k\}_{k=0}^\infty$ *принадлежит множеству* X *и в этой точке выполняются необходимые условия минимума функции* f_0 *на множестве* X.

Алгоритм 1'

Начало. I. Выбрать начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Выбрать константы $\delta_0 > 0$ и $\rho > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Положить $x = x^k$.

V. Положить $\delta = \delta_0$.

VI. Найти множество индексов $\mathcal{J}_\delta(x)$ (шаг VI алгоритма 1).

VII. Найти вектор $h(x)$ (шаг VII алгоритма 1).

VIII. Если $h(x) = 0$, то положить $x^* = x$ и прекратить вычисления; иначе перейти к шагу IX.

IX. Вычислить следующее приближение

$$x^{k+1} = x + \rho h(x).$$

X. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1'. *Пусть* x^* — *решение задачи 1 и выполняются следующие условия:* (*i*) — *для любого достаточно малого* $\delta > 0$ *вспомогательная задача квадратичного программирования — найти* $\arg\min_h \left((\nabla f_0(x),\ h) + \frac{1}{2}\|h\|^2\right)$ *при ограничениях*

$$(\nabla f_j(x),\ h) + f_j(x) \leqslant 0, \quad j \in \mathcal{J}_\delta(x),$$

где

$$\mathcal{J}_\delta(x) = \{j \mid f_j(x) \geqslant \max_{i \in \mathcal{J}} f_i(x) - \delta, \quad j \in \mathcal{J}\},$$

разрешима; (*ii*) — *функции* f_j — *дважды непрерывно дифференцируемы и градиенты* $\nabla f_j(x^*)$, $j \in \mathcal{J}_0(x^*)$, *где*

$$\mathcal{J}_0(x^*) = \{j \mid f_j(x^*) = 0, \quad j \in \mathcal{J}\},$$

линейно независимы; (*iii*) — *в точке* x^* *выполняются необходимые условия минимума в форме*

$$\nabla f_0(x^*) + \sum_{j \in \mathcal{J}_0(x^*)} \lambda_j^0 \nabla f_j(x^*) = 0$$

и $\lambda_j^0 > 0$, $j \in \mathcal{J}_0(x^*)$; (*iv*) — *выполняется достаточное условие локального минимума, т. е.*

$$\left(h, \frac{\partial^2 \varphi(x^*, \lambda^0)}{\partial x^2} h\right) > 0$$

для всех $h \neq 0$ *и удовлетворяющих условию*

$$(h, \nabla f_j(x^*)) = 0, \quad j \in \mathcal{J}_0(x^*),$$

где

$$\varphi(x, \lambda) = f_0(x) + \sum_{j \in \mathcal{J}_0(x^*)} \lambda_j f_j(x);$$

вектор λ^0 *состоит из компонент* λ_j^0, $j \in \mathcal{J}_0(x^*)$.

Тогда существуют такая окрестность V *точки* x^* *и константы* $\delta_0 > 0$, $\alpha > 0$, *что бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1′, сходится к точке* x^* *с любого начального приближения* $x^0 \in V$ *с геометрической скоростью, т. е. существует такое число* $0 \leqslant q_1 \leqslant 1$, *что*

$$\|x^k - x^*\| \leqslant \beta_1 (q_1)^k, \quad \beta_1 < \infty,$$

для всех достаточно больших значений k.

Теорема 1″. *Если выполнены все предположения теоремы 1′ и, кроме того, число индексов в множестве* $\mathcal{J}_0(x^*)$ *равно* n *(размерности пространства), то бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1′ при* $\rho = 1$, *сходится в некоторой окрестности точки* x^* *с квадратичной скоростью, т. е.*

$$\|x^{k+1} - x^*\| \leqslant \beta_2 \|x^k - x^*\|^2, \quad \beta_2 < \infty.$$

2. Ограничения типа равенств

Задача 2. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества X, задаваемого соотношением

$$X = \{x \mid f_j(x) = 0, \quad j = 1, \ldots, m, \quad x \in R^n\}.$$

Предположение 2. Функции f_j, $j = 0, 1, \dots, m$ — дважды непрерывно дифференцируемы.

Алгоритм 2

Начало. I. Выбрать начальное приближение x^0 и числа λ_i^0, $i = 1, \dots, m$.

II. Положить $k = 0$.

Основной цикл. III. Если выполняются равенства

$$\nabla f_0(x^k) + \sum_{i=1}^{m} \lambda_i^k \nabla f_i(x^k) = 0;$$

$$f_i(x^k) = 0, \quad i = 1, \dots, m,$$

то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Вычислить производную $\partial\varphi(x^k, \lambda^k)/\partial x$ функции Лагранжа по формуле

$$\partial\varphi(x^k, \lambda^k)/\partial x = \partial f_0(x^k)/\partial x + \sum_{i=1}^{m} \lambda_i^k \partial f_i(x^k)/\partial x.$$

V. Вычислить матрицу вторых производных $\partial^2\varphi(x^k, \lambda^k)/\partial x^2$ функции Лагранжа по формуле

$$\partial^2\varphi(x^k, \lambda^k)/\partial x^2 = \partial^2 f_0(x^k)/\partial x^2 + \sum_{i=1}^{m} \lambda_i^k \partial^2 f_i(x^k)/\partial x^2.$$

VI. Найти вектор h^k и числа δ_i^k, $i = 1, \dots, m$, удовлетворяющие равенствам

$$\frac{\partial^2\varphi(x^k, \lambda^k)}{\partial x^2} h^k + \sum_{i=1}^{m} \delta_i^k \nabla f_i(x^k) + \frac{\partial\varphi(x^k, \lambda^k)}{\partial x} = 0;$$

$$(\nabla f_i(x^k), h^k) + f_i(x^k) = 0, \quad i = 1, \dots, m.$$

VII. Вычислить следующее приближение

$$x^{k+1} = x^k + h^k.$$

VIII. Вычислить

$$\lambda_i^{k+1} = \lambda_i^k + \delta_i^k, \quad i = 1, \dots, m.$$

IX. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполнено предположение 2, а также следующие дополнительные предположения: (i) — вторые производные функций f_j, $j = 0, 1, \dots, m$, удовлетворяют условию Липшица; (ii) — в точке x^*, которая является решением задачи 2, градиенты $\nabla f_j(x^*)$, $j = 1, \dots, m$ — линейно независимы, так что существуют множители Лагранжа λ_i^*, $i = 1, \dots, m$, удовлетворяющие системе*

$$\nabla f_0(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla f_j(x^*) = 0;$$

$$f_j(x^*) = 0, \quad j = 1, \dots, m;$$

(iii) — выполняются условия локального минимума, т. е.

$$\left(y, \frac{\partial^2 \varphi (x^*, \lambda^*)}{\partial x^2} y\right) > 0, \text{ если } y \neq 0;$$

$$(\nabla f_j (x^*), y) = 0, \quad j = 1, \ldots, m,$$

$$\left(\text{здесь } \varphi (x, \lambda) = f_0 (x) + \sum_{j=1}^{m} \lambda_j f_j (x)\right).$$

Тогда, если начальное приближение x^0 *и числа* λ_i^0, $i = 1, \ldots, m$, *достаточно близки к* x^* *и* λ_i^*, $i = 1, \ldots, m$, *соответственно, то последовательность* $\{x^k; \lambda_i^k, i = 1, \ldots, m\}_{k=0}^{\infty}$, *порожденная алгоритмом 2, сходится к* $(x^*; \lambda_i^*, i = 1, \ldots, m)$ *с квадратичной скоростью.*

3. Ограничения смешанного типа

Задача 3. Найти $\arg\min_{x \in X} f_0 (x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X = \{x \mid f_j (x) \leqslant 0, \quad j \in \mathcal{I}, \quad x \in Y\},$$

где Y — заданное выпуклое замкнутое множество;
$f_j : R^n \to R^1$ $(j \in \mathcal{I})$ — заданные непрерывные функции.

Предположение 3. (*i*) — функции f_j, $j \in \mathcal{I} \cup \{0\}$ — непрерывно дифференцируемы.

Алгоритм 3

Все шаги, за исключением I и VII, такие, как и в алгоритме 1. Шаг I в алгоритме 1 заменить на I′.

I′. Выбрать начальное приближение $x^0 \in Y$.

Шаг VII в алгоритме 1 заменить на VII′.

VII′. Найти решение $h = h (x)$ задачи квадратичного программирования:

$$\text{найти } \arg\min_h \left((\nabla f_0 (x), h) + \frac{1}{2} \| h \|^2\right) \text{ при ограничениях}$$

$$(\nabla f_j (x), h) + f_j (x) \leqslant 0, \quad j \in \mathcal{I}_\delta (x), \quad x + h \in Y.$$

Теорема 3. *Если выполнены предположения 3 и существуют такие константы* $\delta > 0$ и $\alpha > 0$, *что:* (*iii*) — *множество*

$$X_\alpha = \{x \mid f_0 (x) + \alpha \max_{j \in \mathcal{I}} f_j (x) \leqslant f_0 (x^0) + \alpha \max_{j \in \mathcal{I}} f_j (x^0), \ x \in Y\},$$

ограничено; (*iv*) — *градиенты функций* f_j, $j \in \mathcal{I} \cup \{0\}$, в X_α *удовлетворяют условию Липшица, т. е.*

$$\| \nabla f_j (x) - \nabla f_j (\bar{x}) \| \leqslant \gamma \| x - \bar{x} \|, \quad \gamma < \infty, \ x, \ \bar{x} \in X_\alpha;$$

(*v*) — *задача квадратичного программирования*

$$\text{найти } \arg\min_h \left((\nabla f_0 (x), h) + \frac{1}{2} \| h \|^2\right) \tag{5.62}$$

при ограничениях

$$(\nabla f_j(x),\ h) + f_j(x) \leqslant 0, \quad j \in \mathcal{I}_\delta(x), \quad x + h \in Y, \tag{5.63}$$

разрешима относительно h при любых $x \in X_\alpha$ и существуют такие множители Лагранжа $\lambda_j(x)$, $j \in \mathcal{I}_\delta(x)$, что

$$\sum_{j \in \mathcal{I}_\delta(x)} \lambda_j(x) \leqslant \alpha;$$

(*vi*) — *среди функций f_j, $j \in \mathcal{I}$, имеется функция f_j, равная тождественно нулю ($f_j(x) \equiv 0$), то любая предельная точка бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, принадлежит множеству X и в этой точке выполняются необходимые условия минимума функции f_0 на множестве X и, кроме того, справедливо*

$$\max_{j \in \mathcal{I}} f_j(x^k) \to 0 \text{ при } k \to \infty,$$

$$x^k \in Y, \quad k = 0, 1, \ldots .$$

(Множители Лагранжа для задачи (5.62) — (5.63) — это такие неотрицательные числа, которые удовлетворяют неравенству

$$(\nabla f_0(x),\ h(x)) + \|h(x)\|^2 + \sum_{j \in \mathcal{I}_\delta(x)} \lambda_j(x)\,[(\nabla f_j(x),\ h(x)) + f_j(x)] \leqslant$$
$$\leqslant (\nabla f_0(x),\ h) + (h(x),\ h) + \sum_{j \in \mathcal{I}_\delta(x)} \lambda_j(x)\,[(\nabla f_j(x),\ h) + f_j(x)]$$

для всех h, удовлетворяющих условию $x + h \in Y$ и равенствам

$$\lambda_j(x)\,[(\nabla f_j(x),\ h(x)) + f_j(x)] = 0, \quad j \in \mathcal{I}_\delta(x)).$$

4. Метод линеаризации, практически реализуемый на ЭВМ

Задача 4. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\} \cap \{x \mid f_j(x) = 0,$$
$$j = m+1, \ldots, m+l, \quad x \in \boldsymbol{R}^n\},$$

определенного с помощью заданных непрерывно дифференцируемых функций $f_j : \boldsymbol{R}^n \to \boldsymbol{R}^1$.

Данный алгоритм не требует априорного знания констант α и δ, обеспечивающих сходимость алгоритма. Эти константы вычисляются в процессе счета. Для реализации данного алгоритма требуется эффективная стандартная подпрограмма решения задачи квадратичного программирования.

Алгоритм 4

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Выбрать константы $\delta_0 > 0$, $0 < \varepsilon < 1$, α_0 (α_0 — достаточно большое число).

III. Положить $k = 0$.

Основной цикл. IV. Положить $\delta = \delta_k$, $x = x^k$, $\alpha = \alpha_k$.

V. Вычислить

$$\psi(x) = \max\{0, f_1(x), \ldots, f_m(x), |f_{m+1}(x)|, \ldots, |f_{m+l}(x)|\}.$$

VI. Найти множество индексов

$$\mathcal{J}_\delta^-(x) = \{j \mid f_j(x) \geqslant \psi(x) - \delta, \quad j = 1, \ldots, m\};$$

$$\mathcal{J}_\delta^0(x) = \{j \mid f_j(x) \geqslant \psi(x) - \delta, \quad j = m+1, \ldots, m+l\}.$$

VII. Если задача квадратичного программирования найти $\arg\min\limits_h \left((\nabla f_0(x), h) + \frac{1}{2}\|h\|^2\right)$ при ограничениях

$$(\nabla f_j(x), h) + f_j(x) \leqslant 0, \quad j \in \mathcal{J}_\delta^-(x);$$

$$(\nabla f_j(x), h) + f_j(x) = 0, \quad j \in \mathcal{J}_\delta^0(x),$$

совместна, то найти ее решение $h = h^k$, множители Лагранжа λ_j, $j \in \mathcal{J}_\delta^-(x) \cup \mathcal{J}_\delta^0(x)$ и перейти к шагу IX; иначе перейти к шагу VIII.

VIII. Положить $x^{k+1} = x$, $\alpha_{k+1} = \alpha$, $\delta_{k+1} = \delta/2$ и перейти к шагу XVI.

IX. Положить $s = 0$.

X. Вычислить $\rho_k = (^1/_2)^s$.

XI. Если выполняется неравенство

$$f_0(x + \rho_k h^k) + \alpha\psi(x + \rho_k h^k) \leqslant f_0(x) + \alpha\psi(x) - \varepsilon\rho_k\|h^k\|^2,$$

то перейти к шагу XII; иначе положить $s = s + 1$ и перейти к шагу X.

XII. Вычислить следующее приближение

$$x^{k+1} = x + \rho_k h^k.$$

XIII. Положить $\delta_{k+1} = \delta$.

XIV. Если выполняется неравенство

$$\alpha > \sum_{j \in \mathcal{J}_\delta^-(x)} \lambda_j + \sum_{j \in \mathcal{J}_\delta^0(x)} |\lambda_j|,$$

то положить $\alpha_{k+1} = \alpha$ и перейти к шагу XVI; иначе перейти к шагу XV.

XV. Вычислить

$$\alpha_{k+1} = 2\left(\sum_{j \in \mathcal{J}_\delta^-(x)} \lambda_j + \sum_{j \in \mathcal{J}_\delta^0(x)} |\lambda_j|\right).$$

XVI. Положить $k = k + 1$ и перейти к шагу IV.

Для алгоритма 4 справедливы теоремы сходимости, аналогичные теоремам пункта 1. Отметим, что начиная с некоторого значения k, величины α_k и δ_k не будут меняться.

5. Аналог метода линеаризации в детерминированных задачах минимизации почти дифференцируемых функций

Задача 5. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества $X \subset R^n$.

Предположения 5. (*i*) — функция f_0 — почти дифференцируема; (*ii*) — X — выпуклое, ограниченное и замкнутое множество, образованное неравенствами

$$f_j(x) \leqslant 0, \quad j = 1, \ldots, m,$$

где $f_j : R^n \to R^1$ — выпуклые функции.

Данные методы используют общую идею метода линеаризации и основаны на понятии почти градиента минимизируемой функции.

Алгоритм 5

Начало. I. Выбрать: начальное приближение $x^0 \in X$, начальное значение $z^0 \in R^n$, начальное значение шаговых множителей ρ_0, α_0 и величину смещения δ_0, удовлетворяющие условиям теоремы 5.

II. Положить $k = 0$.

Основной цикл. III. Вычислить $\bar{x}_i^k$, $i = 1, \ldots, n$ — независимые случайные величины, равномерно распределенные на отрезках

$$\left[x_i^k - \frac{\delta_k}{2},\ x_i^k + \frac{\delta_k}{2}\right], \quad i = 1, \ldots, n.$$

IV. Вычислить вектор

$$\theta(x^k, k) = \frac{1}{\delta_k} \sum_{i=1}^{n} \left[f_0\left(\bar{x}_1^k, \ldots, x_i^k + \frac{\delta_k}{2}, \ldots, \bar{x}_n^k\right) - f_0\left(\bar{x}_1^k, \ldots, x_i^k - \frac{\delta_k}{2}, \ldots, \bar{x}_n^k\right)\right] e^i,$$

где e^i, $i = 1, \ldots, n$ — i-й орт.

V. Вычислить вектор $y^k \in X$, удовлетворяющий условию

$$(z^k, y^k) = \min_{x \in X} (z^k, x).$$

VI. Вычислить вектор

$$x^{k+1} = x^k - \rho_k (x^k - y^k).$$

VII. Вычислить вектор

$$z^{k+1} = z^k + \alpha_k (\theta(x^k, k) - z^k).$$

VIII. Найти значения шаговых множителей ρ_{k+1}, α_{k+1} и значение смещения δ_{k+1}, удовлетворяющие условиям теоремы 5.

IX. Положить $k = k + 1$ и перейти к шагу III.

Теорема 5. Пусть выполнены предположения 5 и, кроме того, имеют место условия

$$0 \leqslant \rho_k \leqslant 1 \text{ и } \delta_k > 0 \text{ при } k = 0, 1, \ldots;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \rho_k/\alpha_k\delta_k \to 0 \text{ при } k \to \infty;$$

$$\sum_{k=0}^{\infty} (\rho_k/\delta_k)^2 < \infty, \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty;$$

$$|\delta_k - \delta_{k+1}|/\rho_k \to 0, \quad \delta_k \to 0 \text{ при } k \to \infty.$$

Тогда с вероятностью 1 предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 5, принадлежат множеству X^* решений задачи 5 и последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ почти наверное сходится.

Замечание 5. Если существует конструктивный способ вычисления почти градиента функции f_0 в каждой точке x, то на шаге IV алгоритма 5 следует вычислить $g(\bar{x}^k)$ — почти градиент функции f_0 в точке $\bar{x}^k$, и на шаге VII вектор z^{k+1} вычислять по формуле

$$z^{k+1} = z^k + \alpha_k (g(\bar{x}^k) - z^k).$$

6. Аналог метода линеаризации в стохастических задачах минимизации почти дифференцируемых функций

З а д а ч а 6. Найти $\arg\min\limits_{x \in X} EF_0(x, \omega)$ для заданной функции $F_0 : X \times \Omega \to \boldsymbol{R}^1$ и заданного множества $X \subset \boldsymbol{R}^n$ ($EF_0(x, \omega) \triangleq \triangleq \int\limits_{\Omega} F_0(x, \omega)\, \mu(d\omega)$; мера μ может быть неизвестной).

Предположения 6. (*i*) — функция $F_0(x, \omega)$ в некоторой области $X' \supset X$ удовлетворяет по x условию Липшица с константой $\gamma(\omega)$; (*ii*) — X — выпуклое, ограниченное и замкнутое множество, образованное неравенствами

$$f_j(x) \leqslant 0, \quad j = 1, \ldots, m,$$

где $f_j : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $j = 1, \ldots, m$ — выпуклые функции.

Алгоритм 6

Н а ч а л о. I. Задать: начальные приближения $x^0 \in X$, $z^0 \in \boldsymbol{R}^n$, начальные значения шаговых множителей $\rho_1(0)$, $\rho_2(0)$ и величину смещения δ_0, удовлетворяющие условиям теоремы 6.

II. Положить $k = 0$.

О с н о в н о й ц и к л. III. Вычислить $\bar{x}_i^k$, $i = 1, \ldots, n$ — независимые случайные величины, равномерно распределенные на отрезках

$$\left[x_i^k - \frac{\delta_k}{2},\ x_i^k + \frac{\delta_k}{2}\right], \quad i = 1, \ldots, n.$$

IV. Вычислить вектор

$$\xi(x^k, k) = \frac{1}{\delta_k} \sum_{i=1}^{n} \left[F_0 \left(\bar{x}_1^k, \ldots, x_i^k + \frac{\delta_k}{2}, \ldots, \bar{x}_n^k, \omega^k \right) - \right.$$
$$\left. - F_0 \left(\bar{x}_1^k, \ldots, x_i^k - \frac{\delta_k}{2}, \ldots, \bar{x}_n^k, \omega^k \right) \right] e^i,$$

где ω^k — независимые по k наблюдения параметра ω, e^i — i-й орт.

V. Вычислить вектор $y^k \in X$, удовлетворяющий условию

$$(z^k, y^k) = \min_{x \in X} (z^k, x).$$

VI. Вычислить вектор

$$x^{k+1} = x^k - \rho_1(k)(x^k - y^k).$$

VII. Вычислить вектор

$$z^{k+1} = z^k + \rho_2(k)(\xi(x^k, k) - z^k).$$

VIII. Вычислить значения шаговых множителей $\rho_1(k+1)$, $\rho_2(k+1)$ и смещение δ_{k+1}, удовлетворяющие условиям теоремы 6.

IX. Положить $k = k + 1$ и перейти к шагу III.

Теорема 6. *Пусть выполнены предположения 6 и, кроме того, имеют место условия*

$$E\gamma^2(\omega) < \infty;$$
$$0 \leqslant \rho_1(k) \leqslant 1, \quad \delta_k > 0 \quad \textit{при} \quad k = 0, 1, \ldots;$$
$$\sum_{k=0}^{\infty} \rho_1(k) = \infty, \quad \rho_1(k)/\rho_2(k)\,\delta_k \to 0 \quad \textit{при} \quad k \to \infty;$$
$$\sum_{k=0}^{\infty} (\rho_1(k)/\delta_k)^2 < \infty, \quad \sum_{k=0}^{\infty} \rho_2^2(k) < \infty;$$
$$|\delta_k - \delta_{k+1}|/\rho_1(k) \to 0 \quad \textit{и} \quad \delta_k \to 0 \quad \textit{при} \quad k \to \infty.$$

Тогда с вероятностью 1 предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 6, принадлежат множеству X^* решений задачи 6 и последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ почти наверное сходится.

7. Стохастический метод линеаризации

Задача 7. Найти $\arg\min_{x \in X} f_0(x)$, $f_0 : R^n \to R^1$, $X \subset R^n$.

Предположения 7. (*i*) — функция f_0 — непрерывно дифференцируемая; (*ii*) — множество $X \subset R^n$ — ограничено и замкнуто.

Если множество X имеет не простой вид (например, X не является n-мерным параллелепипедом), то решение задачи 7 с использованием операции проектирования на область X требует эффективных способов решения задачи минимизации суммы квадратов при наличии ограничений, что не всегда возможно.

В стохастическом методе линеаризации обычная операция проектирования на множество X заменяется операцией (иногда более простой, чем проектирование) минимизации на X линейной функции (z^k, x), определяемой некоторым вспомогательным вектором z^k.

Обозначим через Z выпуклое, ограниченное и замкнутое множество, для которого

$$Z \supseteq \{\nabla f_0(x) \mid x \in X^*\},$$

где X^* — множество решений задачи 7.

Алгоритм 7

Начало. I. Выбрать произвольные начальные точки $x^0 \in X$ и $z^0 \in Z$.

II. Задать правила построения последовательностей шаговых множителей

$$\{\rho_k\}_{k=0}^{\infty} \quad \text{и} \quad \{\delta_k\}_{k=0}^{\infty}.$$

III. Положить $k = 0$.

Основной цикл. IV. Вычислить вектор $\bar{x}^k$, удовлетворяющий условию

$$(z^k, \bar{x}^k) = \min_{x \in X} (z^k, x).$$

V. Вычислить вектор

$$x^{k+1} = x^k + \rho_k (\bar{x}^k - x^k).$$

VI. Найти случайный вектор ξ^k, условное математическое ожидание которого

$$E(\xi^k/(x^0, z^0), \ldots, (x^k, z^k)) = \nabla f_0(x^k) + b^k,$$

где b^k — вектор, измеримый относительно σ-алгебры, индуцированной величинами $(x^0, z^0), \ldots, (x^k, z^k)$.

Способы построения вектора ξ^k приведены в примерах к § 5.28.

VII. Вычислить вектор

$$z^{k+1} = \pi_Z (z^k + \delta_k (\xi^k - z^k)).$$

VIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 7. *Пусть имеют место предположения 7 и пусть, кроме того, $\nabla f_0(x)$ удовлетворяет локальному условию Липшица. Тогда, если выполнены условия*

$$\|\xi^k\| + \|\nabla f_0(x^k)\| + \|b^k\| \leqslant c_1, \quad c_1 < \infty;$$

$$\rho_k \geqslant 0, \quad \delta_k \geqslant 0, \quad \rho_k/\delta_k \to 0 \quad \text{при} \quad k \to \infty;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \delta_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k \|b^k\| < \infty \ \text{п. н};$$

$$\sum_{k=0}^{\infty} E(\rho_k^2 + \delta_k^2) < \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 7, такова, что* $\{f_0(x^k)\}_{k=0}^{\infty}$ *сходится п. н и каждая предельная точка* $\{x^k\}_{k=0}^{\infty}$ *принадлежит множеству*

$$X^* = \{x^* \mid \min_{x \in X} (\nabla f_0(x^*),\ x - x^*) = 0\}.$$

Библиографические указания. Пункты 1, 2, 3, 4 написаны на основании работ [320, 315, 293], пункты 5, 6 — на результатах работы [155], пункт 7 — на основании работ [153, 99, 214].

5.12. Методы линеаризации в предельных экстремальных задачах

1. Детерминированный случай

Задача 1. Найти $\arg\min_{x \in X} f_0(x)$, где $f_0(x) \triangleq \lim_{k \to \infty} f_0^k(x)$; $f_0^k : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $k = 0, 1, \ldots$ — заданные функции, $X \subset \boldsymbol{R}^n$.

Предположения 1. (*i*) — функции f_0^k, $k = 0, 1, \ldots$, имеют непрерывные производные; (*ii*) — X — выпуклое, ограниченное, замкнутое множество.

Сущность метода состоит в том, что для минимизации функции f_0 в k-й итерации используется вектор $\nabla f_0^k(x^k)$ — градиент функции f_0^k, причем операция проектирования на множество X заменена задачей минимизации линейного функционала $(\nabla f_0^k(x^k),\ x)$ по x на множестве X. Особенность данного метода состоит в том, что сама минимизируемая функция f_0 может и не иметь непрерывных производных.

Определение 1. Множеством решений задачи 1 называется множество X^*, элементы которого обладают следующим свойством: $x^* \in X^*$, если существует последовательность точек $y^k \to x^*$ при $k \to \infty$ такая, что

$$\min_{x \in X} (\nabla f_0^k(y^k),\ x - y^k) \to 0 \quad \text{при} \quad k \to \infty.$$

Если функция f_0 выпуклая вниз и последовательность выпуклых вниз функций f_0^k удовлетворяет соотношению

$$f_0^k(x) \to f_0(x), \quad k \to \infty$$

для каждой точки $x \in X$, то множество X^*, определенное выше, состоит из точек минимума функции $f_0(\cdot)$.

Алгоритм 1

Начало. I. Найти произвольную точку x^0, принадлежащую множеству X.

II. Задать ρ_0.

III. Положить $k = 0$.

Основной цикл. IV. Найти вектор $\bar{x}^k \in X$ — решение задачи минимизации линейной функции $(\nabla f_0^k(x^k),\ x)$ на множестве

X, т. е.

$$(\nabla f_0^k(x^k),\ \bar{x}^k) = \min_{x \in X} (\nabla f_0^k(x^k),\ x).$$

V. Вычислить вектор

$$x^{k+1} = x^k - \rho_k (x^k - \bar{x}^k).$$

VI. Вычислить шаговый множитель ρ_{k+1}, удовлетворяющий условиям теоремы 1.

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Пусть выполнены предположения 1 и, кроме того, имеют место условия:* (*i*) — *последовательность функций* f_0^k, $k = 0, 1, \ldots,$ *сходится к функции* f_0 *равномерно в области* X;

(*ii*) —

$$|f_0^{k+1}(x) - f_0^k(x)| \leqslant \alpha_1 \sigma_k,$$

где $\alpha_1 = \mathrm{const}$, $\sigma_k \to +0$ *при* $k \to \infty$, $x \in X$;

(*iii*) —

$$\|\nabla f_0^k(x) - \nabla f_0^k(y)\| \leqslant \alpha_2 \|x - y\| / \Delta_k,$$

где $\alpha_2 = \mathrm{const}$, $\Delta_k \to +0$ *при* $k \to \infty$, $x, y \in X$. *Тогда, если шаговые множители* ρ_k, $k = 0, 1, \ldots,$ *выбирать такими, что*

$$\rho_k \to +0, \quad \sigma_k/\rho_k \to 0, \quad \rho_k/\Delta_k \to 0 \quad \textit{при} \quad k \to \infty$$

и $\sum_{k=0}^{\infty} \rho_k = \infty$, *то каждая предельная точка последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, принадлежит множеству решений* X^* *задачи 1 и последовательность* $\{f_0(x^k)\}_{k=0}^{\infty}$ *сходится.*

Теорема 1'. *Пусть выполнены предположения 1, условия* (*i*), (*ii*) *теоремы 1 и, кроме того, имеют место условия:* (*i'*) — *функция* f_0 — *непрерывно дифференцируема;*

(*ii'*) —

$$\|\nabla f_0^k(x) - \nabla f_0^k(y)\| \leqslant \beta_k \|x - y\|,$$

где константы β_k, $k = 0, 1, \ldots,$ *ограничены в совокупности;* $x, y \in X$.

Тогда, если шаговые множители ρ_k, $k = 0, 1, \ldots,$ *выбирать такими, что*

$$\rho_k \to +0, \quad \sigma_k/\rho_k \to 0 \quad \text{при} \quad k \to \infty;$$

$$\sum_{k=0}^{\infty} \rho_k = \infty,$$

то каждая предельная точка последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, принадлежит множеству решений* X^* *задачи 1 и последовательность* $\{f_0(x^k)\}_{k=0}^{\infty}$ *сходится.*

Замечание 1. Для выпуклых в ограниченной области X функций f_0, f_0^k, $k = 0, 1, \ldots$, теорема 1 и теорема 1′ остаются в силе и в том случае, когда последовательность функций $f_0^k(x) \to f_0(x)$ при $k \to \infty$ поточечно.

2. Стохастический случай

Задача 2. Найти

$$\arg\min_{x \in X} \lim_{k \to \infty} EF^k(x, \omega)$$

для заданной последовательности функций $\{F^k\}_{k=0}^{\infty}$, $F^k : X \times \Omega \to R^1$ и заданного множества $X \subset R^n$.

Предположения 2. (*i*) — функции $f_0^k(x) \triangleq EF^k(x, \omega)$, $k = 0, 1, \ldots$ — непрерывно дифференцируемы; (*ii*) — X — выпуклое, ограниченное, замкнутое множество.

Сущность метода заключается в том, что для минимизации функции $f_0(x) \triangleq \lim_{k \to \infty} f_0^k(x)$ на k-й итерации используются стохастические квазиградиенты функций f_0^k, т. е. векторы ξ^k, условное математическое ожидание которых

$$E(\xi^k / \mathcal{B}_k) = \nabla f_0^k(x^k) + b^k,$$

где b^k — векторы, измеримые относительно σ-подалгебры $\mathcal{B}_k$ (в частности, можно положить

$$\xi^k = \nabla_x F^k(x^k, \omega)).$$

В данном методе операция проектирования на множество X заменена задачей минимизации линейной функции (z^k, x) по $x \in X$, где векторы z^k «близки» к градиенту минимизируемой функции.

Алгоритм 2

Начало. I. Найти x^0, z^0 — произвольные точки выпуклого, ограниченного, замкнутого множества X.

II. Задать начальные значения шаговых множителей $\rho_1(0)$ и $\rho_2(0)$.

III. Положить $k = 0$.

Основной цикл. IV. Найти вектор $\bar{x}^k \in X$ — решение задачи минимизации линейной функции (z^k, x) на множестве X, т. е.

$$(z^k, \bar{x}^k) = \min_{x \in X} (z^k, x).$$

V. Вычислить вектор

$$x^{k+1} = x^k - \rho_1(k)(x^k - \bar{x}^k).$$

VI. Найти одну реализацию ξ^k случайного вектора $\bar{\xi}^k$, условное математическое ожидание которого

$$E(\bar{\xi}^k / x^0, z^0, \ldots, x^k, z^k) = \nabla f_0^k(x^k) + b^k,$$

где случайный вектор b^k предполагается измеримым относительно σ-подалгебры, индуцированной величинами $(x^0, z^0, \ldots, x^k, z^k)$.

VII. Найти вектор

$$z^{k+1} = z^k - \rho_2(k)(z^k - \xi^k).$$

VIII. Найти шаговые множители $\rho_1(k+1)$ и $\rho_2(k+1)$, удовлетворяющие условиям теоремы 2.

IX. Положить $k = k+1$ и перейти к шагу IV.

Теорема 2. *Пусть выполнены предположения 2 и, кроме того, имеют место условия:* (***i***) *— последовательность функций $f_0^k(x)$ сходится к функции $f_0(x)$ равномерно в области X;*

(***ii***) —.

$$|f_0^{k+1}(x) - f_0^k(x)| \leqslant \chi_1\sigma_k, \quad x \in X, \quad \chi_1 < \infty, \quad \sigma_k \to +0 \text{ при } k \to \infty;$$

(***iii***) —

$$\|\nabla f_0^k(x) - \nabla f_0^k(y)\| \leqslant \chi_2 \|x - y\|/\Delta_k, \quad \chi_2 < \infty, \quad x, y \in X;$$

$$\Delta_k \to +0 \text{ при } k \to \infty; \ (iv) \text{ — для всех } x \in X, \ k = 0, 1, \ldots,$$

$$\|\nabla f_0^{k+1}(x) - \nabla f_0^k(x)\| \leqslant \chi_3\delta_k,$$

где $\chi_3 < \infty$, $\delta_k \to 0$ *при* $k \to \infty$;

(***v***) —

$$\|\bar{\xi}^k\| + \|\nabla f_0^k(x^k)\| + \|b^k\| \leqslant \chi_4 \text{ п. н}, \ k = 0, 1, \ldots,$$

где $\chi_4 < \infty$.

Тогда, если шаговые множители $\rho_1(k)$ и $\rho_2(k)$ *выбирать такими, что*

$$\rho_l(k) \to +0, \quad \sum_{k=0}^{\infty} \rho_l(k) = \infty, \quad l = 1, 2;$$

$$\sum_{k=0}^{\infty} (\rho_2(k))^2 < \infty, \quad \sum_{k=0}^{\infty} \rho_2(k) \|b^k\| < \infty \text{ п. н};$$

$$\rho_1(k)/\rho_2(k)\Delta_k \to 0, \quad \delta_k/\rho_2(k) \to 0,$$

то $\|z^k - \nabla f_0^k(x^k)\| \to 0$ п. н.

Если дополнительно выполняется условие

$$\sigma_k/\rho_1(k) \to 0 \quad \text{при} \quad k \to \infty,$$

то последовательность $\{f_0(x^k)\}$ почти наверное сходится и каждая предельная точка последовательности $\{x^k(\omega)\}$, генерируемой алгоритмом 2, почти наверное принадлежит множеству X^*, определенному согласно определению 1.

Замечание 2. Если функция $f_0(x)$ гладкая, то для справедливости теоремы 2 вместо условия (***iii***) достаточно потребовать выполнения условия

$$\|\nabla f_0^k(x) - \nabla f_0^k(y)\| \leqslant \beta_k \|x - y\|,$$

где β_k, $k = 0, 1, \ldots,$ — ограниченные в совокупности константы.

Тогда надо выбирать $\rho_1(k)$ и $\rho_2(k)$ такими, чтобы

$$\rho_1(k)/\rho_2(k) \to 0 \quad \text{при} \quad k \to \infty.$$

Если же, кроме того, $\lim_{k\to\infty} \nabla f_0^k(x) = \nabla f_0(x)$ для каждого $x \in X$, то теорема 2 справедлива и в том случае, когда σ_k и δ_k при $k \to \infty$ стремятся к нулю произвольно, т. е. не требуется, чтобы

$$\delta_k/\rho_2(k) \to 0 \quad \text{и} \quad \sigma_k/\rho_1(k) \to 0 \quad \text{при} \quad k \to \infty.$$

Библиографические указания. При написании параграфа использовались работы [154, 97, 197].

5.13. Методы отсечения

1. Линейный случай

Задача 1. Найти $\arg\min_{x\in X}(c, x)$ для заданного вектора $c \in R^n$ и множества

$$X \triangleq \{x \mid f_1(x) \leqslant 0, \quad x \in R^n\},$$

определенного с помощью заданной выпуклой функции $f_1: R^n \to R^1$.

Заметим, что более общая задача выпуклого программирования в пространстве R^n: найти $\arg\min_{x\in X} f_0(x)$, где

$$X \triangleq \{x \mid \hat{f}_j(x) \leqslant 0, \quad j = 1, \ldots, m\}, \quad \hat{f}_j, \quad j = 0, 1, \ldots, m,$$

— выпуклые функции, легко приводится к виду задачи 1 с помощью дополнительной координаты x_{n+1} и неравенства

$$\bar{f}_{m+1}(x, x_{n+1}) \triangleq f_0(x) - x_{n+1} \leqslant 0:$$

найти $\arg\min_{(x, x_{n+1})\in \bar{X}} x_{n+1}$,

где

$$\bar{X} = \{(x, x_{n+1}) \mid \bar{f}(x, x_{n+1}) \leqslant 0, \ (x, x_{n+1}) \in R^{n+1}\};$$

$$\bar{f}(x, x_{n+1}) \triangleq \max_{1\leqslant i\leqslant m+1} \bar{f}_i(x, x_{n+1}),$$

$$\bar{f}_i(x, x_{n+1}) \triangleq \hat{f}_i(x) + 0 \cdot x_{n+1}, \quad i = 1, \ldots, m.$$

Предположение 1. Множество X — непустое компактное множество.

Сущность метода отсекающей гиперплоскости заключается в решении задачи 1 путем решения последовательности специальных задач линейного программирования, полученных аппроксимацией множества ограничений X последовательностью многогранников X_k, $k = 0, 1, \ldots$, каждый из которых содержит X.

Алгоритм 1

Начало. I. Выбрать целое число $l > 0$, n-мерные векторы a^i, $i = -l, -(l-1), \dots, -1, 0$ и числа β_i, $i = -l, -(l-1), \dots$ $\dots, -1, 0$, такие, что множество

$$X_0 \triangleq \{x \mid (a^i, x) - \beta_i \leqslant 0, \quad i = -l, \dots, 0, \quad x \in \boldsymbol{R}^n\},$$

компактно и содержит X.

II. Положить $k = 0$.

Основной цикл. III. Найти решение $x = x^k$ задачи линейного программирования:

найти $\arg\min\limits_x (c, x)$ при ограничениях

$$(a^i, x) - \beta_i \leqslant 0, \quad i = -l, \quad -(l-1), \dots, k. \qquad (5.64)$$

IV. Если $f_1(x^k) \leqslant 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу V.

V. Вычислить опорный вектор a^{k+1} к функции $f_1(x)$ в точке x^k.

VI. Вычислить

$$\beta_{k+1} = (a^{k+1}, x^k) - f_1(x^k).$$

VII. Построить множество

$$X_{k+1} \triangleq \{x \mid (a^{k+1}, x) - \beta_{k+1} \leqslant 0, \quad x \in \boldsymbol{R}^n\} \cap X_k.$$

VIII. Положить $k = k + 1$ и перейти к шагу III.

Замечание 1. На шаге III алгоритма 1 требуется решать задачу линейного программирования, число ограничений которой растет с увеличением k. Это требует большой оперативной памяти ЭВМ для хранения векторов a^i, $i = -l, \dots, k$, и чисел β_i, $i = -l, \dots, k$. Для упрощения алгоритма на шаге III вместо задачи линейного программирования (5.64) можно решать двойственную ей задачу:

$$\text{найти } \arg\max_u \left(-\sum_{i=-l}^{k} u_i \beta_i\right)$$

при ограничениях

$$\sum_{i=-l}^{k} u_i a^i + c = 0, \quad u_i \geqslant 0, \quad i = -l, \dots, k.$$

При решении этих двойственных задач симплекс-методом весьма существенно, что решение предыдущей задачи служит хорошим допустимым решением для последующей, так что их решение получают довольно быстро.

Теорема 1. *Если: (i) — $f_1(x)$ — выпуклая непрерывная функция; (ii) — множество X — непустое и компактное; (iii) — существует такое число $\delta < \infty$, что при каждом $\tilde{x} \in X_0$ вектор a, опорный к $f_1(x)$ в точке $\tilde{x}$, удовлетворяет неравенству $\| a \| \leqslant \delta$, то любая предельная точка x^* бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, является решением задачи 1 и $f_1(x^k) \to 0$ при $k \to \infty$.*

2. Общий случай

З а д а ч а 2. Найти $\arg\min_{x \in X} f_0(x)$ для заданной выпуклой функции $f_0 : R^n \to R^1$ и множества X, задаваемого соотношением

$$X = \{x \mid f_1(x) \leqslant 0, \quad x \in R^n\},$$

где $f_1 : R^n \to R^1$ — непрерывная выпуклая функция.

Алгоритм 2

Н а ч а л о. I. Выбрать выпуклое замкнутое множество X_0, содержащее X.

II. Положить $k = 0$.

О с н о в н о й ц и к л. III. Решить задачу выпуклого программирования

$$x^k = \arg\min_{x \in X_k} f_0(x).$$

IV. Если $f_1(x^k) \leqslant 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу V.

V. Вычислить опорный вектор $\hat{\nabla} f_1(x^k)$ функции f_1 в точке x^k.

VI. Построить множество

$$X_{k+1} \triangleq X_k \cap \{x \mid f_1(x^k) + (\hat{\nabla} f_1(x^k), \quad x - x^k) \leqslant 0, \quad x \in R^n\}.$$

VII. Положить $k = k + 1$ и перейти к шагу III.

Множества X_k строят таким образом, чтобы решение этой задачи было проще, чем решение исходной задачи 2.

Теорема 2. *Если: (i) — $f_0(x)$ — равномерно выпуклая функция на множестве X_0; (ii) — X_0 — выпуклое и замкнутое множество; (iii) — $f_1(x)$ — непрерывная выпуклая функция, удовлетворяющая условию Липшица на X_0, то для бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, справедливо:*

1) $\lim_{k \to \infty} f_0(x^k) \leqslant f_0^* \triangleq \min_{x \in X} f_0(x), \quad \overline{\lim_{k \to \infty}} f_1(x^k) \leqslant 0;$

2) *если, кроме того, $f_1(x)$ — корректное ограничение и $f_0(x)$ удовлетворяет условию Липшица на множестве X_0, то*

$$\lim_{k \to \infty} x^k = x^*, \quad x^* \in X; \quad f_0(x^*) = f_0^*; \quad \lim_{k \to \infty} f_0(x^k) = f_0^*;$$

3) *если, кроме того, $f_0(x)$ — сильно выпуклая функция и существует внутренняя точка $\bar{x}$ множества X_0 такова, что $f_1(\bar{x}) < 0$, то имеют место следующие оценки сходимости:*

$$f_0(x^k) - f_0^* \leqslant \alpha_1 / k, \quad \alpha_1 < \infty;$$

$$x^k - x^* \leqslant \alpha_2 / \sqrt{k}, \quad \alpha_2 < \infty.$$

3. Метод отсечения с растяжением пространства для решения задач выпуклого программирования

Задача 3. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : \mathbf{R}^n \to \mathbf{R}^1$ и заданного множества

$$X \triangleq \{x \mid f_i(x) \leqslant 0, \quad i = 1, \ldots, m, \quad x \in \mathbf{R}^n\}.$$

Предположения 3. (*i*) — функции f_i, $i = 0, 1, \ldots, m$ — непрерывны и выпуклы.

Данный метод отсечения с растяжением пространства гарантирует уменьшение объема области, в которой локализируется оптимум, со скоростью геометрической прогрессии, причем знаменатель прогрессии зависит лишь от размерности задачи. Этот алгоритм относится к классу алгоритмов обобщенного градиентного спуска с растяжением пространства в направлении градиента. Кроме того, его можно рассматривать как метод отсечения, в котором операция растяжения пространства используется для симметризации области, в которой локализован оптимум.

Алгоритм 3

Начало. I. Выбрать начальное приближение $x^0 \in \mathbf{R}^n$ и число $\gamma > 0$, удовлетворяющие условиям теоремы 3; положить $B_0 = I$ (I — единичная матрица), а также

$$\rho_0 = \gamma/(n+1), \quad \beta = \sqrt{\frac{n-1}{n+1}}, \quad k = 0.$$

Основной цикл. II. Вычислить вектор

$$g(x^k) = \begin{cases} g_0(x^k), & \text{если } \max\limits_{1 \leqslant i \leqslant m} f_i(x^k) \leqslant 0; \\ g_{i^*}(x^k), & \text{если } \max\limits_{1 \leqslant i \leqslant m} f_i(x^k) \triangleq f_{i^*}(x^k) > 0, \end{cases}$$

где $g_i(x^k)$, $i = 0, 1, \ldots, m$ — субградиенты (обобщенные градиенты) функций f_i, $i = 0, 1, \ldots, m$ соответственно.

III. Если $g(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Вычислить новое направление ξ^k для растяжения пространства

$$\xi^k = B_k^T g(x^k) / \| B_k^T g(x^k) \|,$$

где B_k^T — матрица, транспонированная к матрице B_k.

V. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k B_k \xi^k.$$

VI. Вычислить матрицу B_{k+1}, которая по существу является обратной к результирующей матрице A_{k+1} растяжения пространства после $(k+1)$-го шага с коэффициентом растяжения $\alpha = 1/\beta$

$$B_{k+1} = B_k G_\beta(\xi^k),$$

где $G_\beta(\xi^k)$ — оператор растяжения пространства в направлении ξ^k с коэффициентом β (определение оператора растяжения пространства приведено в § 3.2).

VII. Вычислить значение шагового множителя

$$\rho_{k+1} = \rho_k \frac{n}{\sqrt{n^2-1}} .$$

VIII. Положить $k = k + 1$ и перейти к шагу II.

Теорема 3. *Если выполнено предположение 3 и существует оптимальная точка x^* (не обязательно единственная), находящаяся в шаре радиусом γ с центром в точке x^0, то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 3, удовлетворяет неравенствам*

$$\|A_k(x^k - x^*)\| \leqslant \rho_k(n+1), \tag{5.65}$$

где $A_k \triangleq B_k^{-1}$, $k = 0, 1, \ldots$.

Замечание 3. Неравенство (5.65) дает полезную геометрическую интерпретацию метода, так как множество точек x, удовлетворяющих неравенству

$$\|A_k(x^k - x)\| \leqslant (n+1)\rho_k = \gamma\left(\frac{n}{\sqrt{n^2-1}}\right)^k,$$

представляет собой эллипсоид Φ_k, объем которого

$$V(\Phi_k) = V_0\gamma^k\left(\frac{n}{\sqrt{n^2-1}}\right)^{nk}/\det A_k,$$

где V_0 — объем единичного n-мерного шара.

В связи с этим объем эллипсоида, в котором локализуется оптимальная точка x^*, убывает со скоростью геометрической прогрессии со знаменателем $q_n = \sqrt{\frac{n-1}{n+1}}\left(\frac{n}{\sqrt{n^2-1}}\right)^n < 1$.

Замечание 3'. К алгоритму 3 приводит также схема последовательных отсечений [417]. Действительно, если рассмотреть шар $S_0 \triangleq \{x \mid (\|x - x^0\|) \leqslant \gamma\}$ и провести гиперплоскость $(g(x^0), x - x^0) = 0$, то область локализации x^*, очевидно, сужается до половины шара $\overline{S} = S_0 \cap \Pi_0$, где $\Pi_0 \triangleq \{x \mid (g(x^0), x - x^0) \leqslant 0\}$.

Если теперь описать вокруг шара $\overline{S}$ эллипсоид минимального объема, то его центр будет

$$x^0 - \frac{\gamma}{n+1}\frac{g(x^0)}{\|g(x^0)\|} = x^0 - \frac{\gamma}{n+1}\xi^0,$$

т. е. будет совпадать с x^1.

Если теперь произвести растяжение пространства в направлении $\xi^0 = \frac{g(x^0)}{\|g(x^0)\|}$, то в растянутом пространстве образом указанного эллипсоида будет шар радиусом $\gamma\frac{n}{\sqrt{n^2-1}}$ с центром в точке $G_\alpha(\xi^0)x^1$ и т. д.

Ниже приводится обобщение алгоритма 3.

Предположение 3'. Пусть $g(x)$ — векторное поле, определенное на $\boldsymbol{R}^n$, такое, что

$$(g(x),\ x-x^*)\geqslant 0,\quad \forall x\in \boldsymbol{R}^n,$$

где x^* — искомая оптимальная точка функции f_0 (причем $g(x)\neq 0$ при $x\neq x^*$).

Приведем пример определения векторного поля $g(z)$. Пусть задана выпукло-вогнутая функция $f(x,\ y)$ двух векторных переменных $x\in \boldsymbol{R}^n$, $y\in \boldsymbol{R}^m$. Обозначим $z=(x,\ y)\in \boldsymbol{R}^n\times \boldsymbol{R}^m \triangleq \boldsymbol{R}^{n+m}$. Сформируем векторное поле $g(z)$ следующим образом:

$$g(z)=\{g_f^x(z),\ -g_f^y(z)\},\quad g_f^x(z)\in G_f^x(z),\quad g_f^y(z)\in G_f^y(z),$$

где $G_f^x(x,\ y)$ — множество частных субградиентов (обобщенных градиентов) функции $f(x,\ y)$, которая рассматривается как функция от x при фиксированном y; $G_f^y(x,\ y)$ — множество частных субградиентов от функции $f(x,\ y)$ по y при фиксированном x.

Алгоритм 3'

Начало. I. Выбрать начальное приближение $x^0\in \boldsymbol{R}^n$, лежащее на расстоянии, не превышающем γ от искомой оптимальной точки x^*, положить $B_0=I$; $\rho=\gamma/(n+1)$; $\beta=\sqrt{\frac{n-1}{n+1}}$, $k=0$.

Основной цикл. II. Вычислить вектор $g(x^k)$ — значение векторного поля, определенного согласно предположению 3', в точке x^k.

III. Если $g(x^k)=0$, то положить $x^*=x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Вычислить вектор ξ^k направления растяжения пространства

$$\xi^k=B_k^T g(x^k)/\|B_k^T g(x^k)\|.$$

V. Вычислить следующее приближение

$$x^{k+1}=x^k-\rho_k B_k \xi^k.$$

VI. Вычислить $n\times n$-матрицу B_{k+1} (обратную результирующей матрице растяжения пространства после $(k+1)$-го шага с коэффициентом растяжения $\alpha=1/\beta$)

$$B_{k+1}=B_k G_\beta(\xi^k),$$

где $G_\beta(\xi^k)$ — оператор растяжения пространства в направлении ξ^k с коэффициентом β.

VII. Вычислить значение шагового множителя

$$\rho_{k+1}=\rho_k\frac{n}{\sqrt{n^2-1}}.$$

VIII. Положить $k=k+1$ и перейти к шагу II.

Теорема 3'. Пусть выполнено предположение 3' и точка x^0 лежит на расстоянии, не превышающем γ от оптимальной точки x^*.

Тогда при $n > 1$ последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 3', удовлетворяет неравенствам

$$\| A_k (x^k - x^*) \| \leqslant \rho_k (n+1), \quad A_k = B_k^{-1}, \quad k = 0, 1, \ldots .$$

Теорема 3' дает оценки сходимости алгоритма 3', выраженные в терминах уменьшения объема области локализации минимума.

Приведенная ниже теорема 3'' характеризует скорость сходимости алгоритма 3' по функционалу.

Теорема 3''. Если функция f_0 выпукла в $\boldsymbol{R}^n$ и в процессе применения алгоритма 3' выполняются следующие условия:

(i) —

$$\| x^0 - x^* \| \leqslant \gamma;$$

(ii) —

$$\| g(x^k) \| \leqslant \sigma, \quad k = 0, 1, \ldots,$$

здесь $g(x^k)$ — обобщенный градиент функции f_0 в точке x^k,

то для последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3', справедливы неравенства

$$\min_{0 \leqslant r \leqslant k} (f_0(x^r) - f_0(x^*)) \leqslant$$

$$\leqslant \gamma \sigma_k \sqrt{k(\alpha^2 - 1)}\, q_n^{k/n} / \sqrt{1 - \beta^{2k/n}}, \quad k = 1, 2, \ldots,$$

где

$$\sigma_k = \max_{0 \leqslant r \leqslant k} \| g(x^r) \|, \quad \alpha = 1/\beta;$$

$$q_n = \sqrt{\frac{n-1}{n+1}} \left(\frac{n}{\sqrt{n^2 - 1}} \right)^n .$$

Замечание 3''. Так как для достаточно больших n $q_n \approx 1 - \frac{1}{2n}$, то теорема 3'' гарантирует скорость сходимости «рекордов отклонения» функционалов от оптимального значения, соответствующую скорости сходимости геометрической прогрессии со знаменателем $q \approx 1 - \frac{1}{2n^2}$.

Библиографические указания. Пункт 1 написан на основании работ [320, 330], пункт 2 основан на результатах работы [222], пункт 3 написан на основании [402, 403, 404, 462, 417].

В работе [443] предлагается вариант метода отсечения для решения задачи вогнутого программирования с линейными ограничениями.

5.14. Методы, использующие функцию Лагранжа

1. Градиентный метод для задач с ограничениями типа неравенств

Задача 1. Найти $\arg\max_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X \triangleq \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m, \quad x \in R^n\}.$$

Предположения 1. (*i*) — функции $f_j(x)$, $j = 0, 1, \ldots, m$,— вогнуты и непрерывно дифференцируемы в R^n ($f_0(x)$, кроме того, строго вогнута); (*ii*) — выполняется условие Слейтера, т. е. существует такая точка $\bar{x}$, что $f_j(\bar{x}) > 0$, $j = 1, \ldots, m$; (*iii*) — существует стационарная точка x^* задачи 1 такая, что

$$f_1(x^*) = 0, \ \ldots, \ f_p(x^*) = 0, \quad f_{p+1}(x^*) > 0, \ \ldots, \ f_m(x^*) > 0;$$

(*iv*) — функции $f_j(x)$, $j = 0, 1, \ldots, m$, имеют в точке x^* вторую производную.

Введем функцию

$$\tilde{\varphi}(x, \tilde{y}) \triangleq f_0(x) + \sum_{i=1}^{p} y_i f_i(x), \quad \tilde{y} = (y_1, \ldots, y_p) \in R_+^p,$$

которая является ограничением функции Лагранжа

$$\varphi(x, y) \triangleq f_0(x) + \sum_{i=1}^{m} y_i f_i(x), \quad y = (y_1, \ldots, y_m) \in R_+^m, \qquad (5.66)$$

на множество $R^n \times R_+^p$.

Поскольку x^* — стационарная точка задачи 1, то существует вектор $\tilde{y}^* = (y_1^*, \ldots, y_p^*) \in R_+^p$ такой, что $\nabla_x \tilde{\varphi}(x^*, \tilde{y}^*) = 0$.

Ниже приводится метод множителей Лагранжа с постоянным шаговым множителем, который при определенных условиях сходится глобально к седловой точке функции Лагранжа со скоростью геометрической прогрессии.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $(x^0, y^0) \in R^n \times R_+^m$.

II. Выбрать постоянный шаговый множитель $\rho > 0$.

III. Положить $k = 0$.

IV. Определить функцию Лагранжа $\varphi(x, y)$ по (5.66).

Основной цикл. V. Вычислить вектор $\nabla_x \varphi(x^k, y^k)$.

VI. Положить $h^k = (f_1(x^k), \ldots, f_m(x^k))$.

VII. Вычислить следующие приближения x^{k+1} и y^{k+1} по формулам

$$x^{k+1} = x^k + \rho \nabla_x \varphi(x^k, y^k);$$
$$y^{k+1} = \max\{0, \ y^k - \rho h^k\}.$$

VIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. Пусть выполняются предположения 1 и пусть: (v) — *задача 1 строго регулярна, т. е.*

1) *не существует ненулевого вектора* $u \in R^n$ *такого, что*

$$-\nabla^2_{xx}\tilde{\varphi}(x^*, \tilde{y}^*)u = 0, \quad \nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*)u = 0;$$

2) *не существует ненулевого вектора* $v \in R^p$ *такого, что*

$$(\nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*))^T v = 0$$

(*т. е., векторы* $\nabla f_1(x^*), \ldots, \nabla f_p(x^*)$ — *линейно независимы);*

3) *выполняется условие строгой дополняющей нежесткости*

$$\tilde{y}^* \in \operatorname{Int} R^p_+, \text{ т. е. } y_i^* > 0, \quad i = 1, \ldots, p;$$

(vi) — *не существует числа* $\lambda \neq 0$ *и ненулевого вектора* $u \in R^n$ *таких, что*

$$-\nabla^2_{xx}\tilde{\varphi}(x^*, \tilde{y}^*)u = 0;$$

$$(\nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*))^T \nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*)u = \lambda^2 u.$$

Тогда для любого $\delta > 0$ *найдется число* $\rho(\delta) > 0$ *такое, что алгоритм 1 при всяком начальном приближении* (x^0, y^0), *удовлетворяющем условию*

$$\|(x^0, y^0) - (x^*, y^*)\| \leqslant \delta,$$

и произвольном постоянном шаговом множителе $\rho \leqslant \rho(\delta)$, *порождает последовательность* $\{(x^k, y^k)\}_{k=0}^{\infty}$, *которая линейно (т. е. со скоростью геометрической прогрессии) сходится к седловой точке* (x^*, y^*) *задачи 1.*

Замечание 1. Условие (vi) теоремы 1 является не только достаточным, но и в некотором смысле необходимым для сходимости алгоритма 1 в строго регулярной задаче.

2. Градиентный метод для задач с ограничениями типа равенств

Задача 2. Найти $\arg\min\limits_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X \triangleq \{x \mid f_i(x) = 0, \quad i = 1, \ldots, m, \quad x \in R^n\}.$$

Предположения 2. (i) — задача 2 имеет решение x^*; (ii) — функции $f_j(x)$, $j = 0, 1, \ldots, m$, непрерывно дифференцируемы в окрестности точки x^*; (iii) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — дважды дифференцируемы в точке x^*; (iv) — векторы $\nabla f_i(x^*)$, $i = 1, \ldots, m$ — линейно независимы.

Предположения 2 гарантируют существование и единственность множителей Лагранжа $y^* = (y_1^*, \ldots, y_m^*)$ задачи 2, т. е.

$$\nabla_x \varphi(x^*, y^*) = 0, \quad \nabla_y \varphi(x^*, y^*) = 0,$$

где

$$\varphi(x, y) \triangleq f_0(x) + \sum_{i=1}^{m} y_i f_i(x)$$

— функция Лагранжа задачи 2.

Введем обозначения $A \triangleq \nabla^2_{xx}\varphi(x^*, y^*)$, $C \triangleq \nabla^2_{yx}\varphi(x^*, y^*)$, C^T — матрица, транспонированная к C.

Приведенный ниже алгоритм порождает последовательность $\{(x^k, y^k)\}_{k=0}^{\infty}$, которая локально (т. е. при начальном приближении (x^0, y^0) из окрестности точки (x^*, y^*)) сходится со скоростью геометрической прогрессии к точке (x^*, y^*). На k-й итерации алгоритма движение к следующему приближению x^{k+1} осуществляется в направлении антиградиента по x функции Лагранжа, а движение к y^{k+1} — в направлении градиента по y этой же функции.

Алгоритм 2

Начало. I. Выбрать начальное приближение (x^0, y^0) из окрестности точки (x^*, y^*).

II. Выбрать постоянный шаговый множитель $\rho > 0$.

III. Определить функцию Лагранжа $\varphi(x, y)$ для задачи 2, т. е. положить

$$\varphi(x, y) \triangleq f_0(x) + \sum_{i=1}^{m} y_i f_i(x), \quad y \in \boldsymbol{R}^m. \tag{5.67}$$

IV. Положить $k = 0$.

Основной цикл. V. Вычислить векторы

$$\nabla_x\varphi(x^k, y^k) = \nabla f_0(x^k) + \sum_{i=1}^{m} y_i^k \nabla f_i(x^k);$$

$$\nabla_y\varphi(x^k, y^k) = (f_1(x^k), \ldots, f_m(x^k)).$$

VI. Вычислить следующие приближения:

$$x^{k+1} = x^k - \rho\nabla_x\varphi(x^k, y^k);$$

$$y^{k+1} = y^k + \rho\nabla_y\varphi(x^k, y^k).$$

VII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 2. *Пусть выполняются предположения 2 и пусть:* (**v**) — *матрица* AC^T *имеет ранг* m, *т. е. из* $AC^Tu = 0$ *следует* $u = 0$; (**vi**) — *матрица* A — *неотрицательно определенная, причем* $Ax \neq 0$ *при* $Cx = 0$, $x \neq 0$.

Тогда существует число $\bar\rho > 0$ *такое, что при* $\rho < \bar\rho$ *алгоритм 2 локально сходится к* (x^*, y^*) *со скоростью геометрической прогрессии (т. е. найдутся числа* $\varepsilon > 0$, $0 \leqslant q_1 < 1$ *такие, что при любых* $\| x^0 - x^* \| \leqslant \varepsilon$, $\| y^0 - y^* \| \leqslant \varepsilon$ *будет*

$$\| x^k - x^* \| \leqslant \beta_1(\varepsilon)(q_1)^k;$$

$$\| y^k - y^* \| \leqslant \beta_1(\varepsilon)(q_1)^k).$$

3. Метод квадратичной аппроксимации для задач с ограничениями типа равенств

В методе квадратичных аппроксимаций на k-й итерации функция Лагранжа $\varphi(x, y)$ квадратично аппроксимируется по x в окрестности точки x^k и следующее приближение x^{k+1} находят из условия минимума этой квадратичной аппроксимации. Движение по переменной y осуществляется, как и в градиентном методе.

Алгоритм 3

Начало. I. Выбрать начальное приближение (x^0, y^0) из окрестности точки (x^*, y^*).

II. Выбрать постоянный шаговый множитель $\rho > 0$.

III. Определить функцию Лагранжа по (5.67).

IV. Положить $k = 0$.

Основной цикл. V. Вычислить векторы

$$\nabla_y \varphi(x^k, y^k) = (f_1(x^k), \ldots, f_m(x^k));$$

$$\nabla_x \varphi(x^k, y^k) = \nabla f_0(x^k) + \sum_{i=1}^{m} y_i^k \nabla f_i(x^k).$$

VI. Вычислить матрицу

$$\nabla^2_{xx} \varphi(x^k, y^k) = \nabla^2_{xx} f_0(x^k) + \sum_{i=1}^{m} y_i^k \nabla^2_{xx} f_i(x^k).$$

VII. Вычислить следующее приближение (x^{k+1}, y^{k+1}) из системы

$$\nabla^2_{xx} \varphi(x^k, y^k)(x^{k+1} - x^k) = -\rho \nabla_x \varphi(x^k, y^k);$$

$$y^{k+1} = y^k + \rho \nabla_y \varphi(x^k, y^k).$$

VIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 3. *Пусть выполняются предположения 2 и пусть (**vii**) — матрица $A \triangleq \nabla^2_{xx}\varphi(x^*, y^*)$ — положительно определенная, т. е. $(Ax, x) > 0$ при $x \neq 0$.*

Тогда существует число $\bar{\rho} > 0$ такое, что при $\rho < \bar{\rho}$ алгоритм 3 локально сходится к (x^*, y^*) со скоростью геометрической прогрессии (т. е. найдутся числа $\varepsilon > 0$, $0 \leqslant q_2 < 1$ такие, что при любых $\| x^0 - x^* \| \leqslant \varepsilon$, $\| y^0 - y^* \| \leqslant \varepsilon$ будет

$$\| x^k - x^* \| \leqslant \beta_2(\varepsilon)(q_2)^k;$$

$$\| y^k - y^* \| \leqslant \beta_2(\varepsilon)(q_2)^k).$$

4. Двойственный метод для задач с ограничениями типа равенств

В двойственном методе на k-й итерации вычисляют $(k+1)$-е приближение x^{k+1} путем решения задачи безусловной минимизации по x функции Лагранжа $\varphi(x, y^k)$ (предполагается, что задача

безусловной минимизации разрешима). Движение по переменной y осуществляется, как и в градиентном методе.

Алгоритм 4

Начало. I. Выбрать начальное приближение (x^0, y^0) из окрестности точки (x^*, y^*).

II. Выбрать постоянный шаговый множитель $\rho > 0$.

III. Определить функцию Лагранжа по (5.67).

IV. Положить $k = 0$.

Основной цикл. V. Вычислить вектор

$$\nabla_y \varphi(x^k, y^k) = (f_1(x^k), \ldots, f_m(x^k)).$$

VI. Вычислить следующее приближение (x^{k+1}, y^{k+1}) по правилам

$$\varphi(x^{k+1}, y^k) = \min_{x \in R^n} \varphi(x, y^k);$$

$$y^{k+1} = y^k + \rho \nabla_y \varphi(x^k, y^k).$$

VII. Положить $k = k + 1$ и перейти к шагу V.

Для алгоритма 4 имеет место теорема, аналогичная теореме 3.

5. Метод Ньютона для задач с ограничениями типа равенств

Предположения 5. Выполняются все условия предположения 2 и, кроме того, (ix) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — дважды дифференцируемы в окрестности точки x^*, причем вторые производные удовлетворяют условию Липшица.

Алгоритм 5

Начало. I. Выбрать начальное приближение (x^0, y^0) из окрестности точки (x^*, y^*).

II. Определить функцию Лагранжа по (5.67).

III. Положить $k = 0$.

Основной цикл. IV. Вычислить векторы

$$\nabla_x \varphi(x^k, y^k) = \nabla f_0(x^k) + \sum_{i=1}^{m} y_i^k \nabla f_i(x^k);$$

$$\nabla_y \varphi(x^k, y^k) = (f_1(x^k), \ldots, f_m(x^k)).$$

V. Вычислить $n \times n$-матрицу $\nabla_{xx}^2 \varphi(x^k, y^k)$ и $n \times m$-матрицу $\nabla_{xy}^2 \varphi(x^k, y^k)$ по формулам

$$\nabla_{xx}^2 \varphi(x^k, y^k) = \nabla_{xx}^2 f_0(x^k) + \sum_{i=1}^{m} y_i^k \nabla_{xx}^2 f_i(x^k);$$

$$\nabla_{xy}^2 \varphi(x^k, y^k) = (\nabla f_1(x^k), \ldots, \nabla f_m(x^k)).$$

VI. Найти решение (x^{k+1}, y^{k+1}) следующей системы линейных уравнений:

$$\nabla_{xx}^2 \varphi(x^k, y^k)(x^{k+1} - x^k) + \nabla_{xy}^2 \varphi(x^k, y^k)(y^{k+1} - y^k) = -\nabla_x \varphi(x^k, y^k);$$

$$(\nabla_{xy}^2 \varphi(x^k, y^k))^T (x^{k+1} - x^k) = -\nabla_y \varphi(x^k, y^k).$$

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 5. *Пусть выполняются предположения 5 и пусть* (x) — *матрица* $A \triangleq \nabla^2_{xx} \varphi(x^*, y^*)$ *удовлетворяет условию* $(Ax, x) \neq 0$ *при* $\nabla^2_{yx}(x^*, y^*)x = 0$, $x \neq 0$. *Тогда алгоритм 5 локально сходится к* (x^*, y^*) *с квадратичной скоростью, т. е.*

$$\|x^k - x^*\| \leqslant \beta_3(\varepsilon)(q_3)^{2^k}, \quad \|y^k - y^*\| \leqslant \beta_3(\varepsilon)(q_3)^{2^k}, \quad q_3 < 1,$$

при

$$\|x^0 - x^*\| \leqslant \varepsilon, \quad \|y^0 - y^*\| \leqslant \varepsilon.$$

Замечание 5. Наиболее простым для реализации является градиентный метод, однако сходимость в алгоритме 2 носит немонотонный характер, поэтому на практике трудно определить — имеет ли место сходимость при выбранном ρ, или нет. Алгоритмы квадратичной аппроксимации и двойственный, как правило, сходятся быстрее градиентного метода, однако они более трудоемки с точки зрения вычислений. Метод Ньютона по трудности вычислений сравним с методами пункта 3 и 4, однако он сходится с квадратичной скоростью и не требует находить шаговый множитель ρ, обеспечивающий сходимость.

Замечание 5'. Для начала работы алгоритмов, приведенных в пунктах 2—5, необходимо иметь начальное приближение из окрестности точки (x^*, y^*). В [293] для получения хорошего начального приближения предполагается использовать метод штрафных функций. В начале необходимо найти хорошее приближение x^k для x^*, минимизируя функцию

$$f_0(x) + \sum_{i=1}^{m} \frac{1}{2} \alpha_k (f_i(x))^2$$

по x на всем пространстве R^n при достаточно больших значениях α_k. Тогда вектор $(\alpha_k f_1(x^k), \ldots, \alpha_k f_m(x^k))$ будет хорошим приближением для y^*.

Библиографические указания. Пункт 1 написан на основании работы [231] пункты 2, 3, 4, 5 основаны на результатах работ [293, 296, 298, 300, 451, 51].

В работе [558] исследуется алгоритм Куна — Таккера решения задачи нелинейного программирования с линейными ограничениями.

5.15. Методы, использующие модифицированные функции Лагранжа

1. Градиентный метод

Задача 1. Найти $\arg\max_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества X,

$$X \triangleq \{x \mid f_j(x) \geqslant 0, \quad j = 1, \ldots, m\}.$$

Предположения 1. (*i*) — функции f_j, $j = 0, 1, \dots, m$ — дифференцируемые, выпуклые и их производные на любом компакте удовлетворяют условию Липшица; (*ii*) — множество седловых точек $Z^* = \{(x^*, u^*)\}$ функции Лагранжа

$$\varphi(x, u) \triangleq f_0(x) + \sum_{j=1}^{m} f_j(x) u_j \tag{5.68}$$

— непустое.

Приводимый метод основан на построении модифицированного лагранжиана $\psi(x, u)$ задачи 1, т. е. функции

$$\psi(x, u) \triangleq f_0(x) - \frac{1}{2\gamma} \sum_{j=1}^{m} (([u_j - \gamma f_j(x)]_+)^2 - u_j^2), \quad \gamma > 0, \tag{5.69}$$

где $[\alpha]_+ = (\alpha + |\alpha|)/2$, множество седловых точек которого совпадает с множеством седловых точек классического лагранжиана $\varphi(x, u)$.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $(x^0, u^0) \in R^n \times R^m$.

II. Выбрать константу $\gamma > 0$.

III. Положить $k = 0$.

IV. Найти произвольное положительное число ρ_0, удовлетворяющее неравенству

$$\rho_0 \leqslant \min\{\gamma, 2^{-1} \| \nabla\psi(x^0, u^0) \|^{-2}\},$$

где $\nabla\psi(x^0, u^0) = (\nabla_x \psi(x^0, u^0), \nabla_u \psi(x^0, u^0))$, а функция $\psi : R^n \times R^m \to R^1$ определяется по (5.69).

Основной цикл. V. Вычислить вектор

$$x^{k+1} = x^k + \rho_k \nabla_x \psi(x^k, u^k).$$

VI. Вычислить вектор

$$u^{k+1} = u^k - \rho_k \nabla_u \psi(x^k, u^k).$$

VII. Вычислить градиент $\nabla\psi(x^{k+1}, u^{k+1})$ функции ψ в точке $\{x^{k+1}, u^{k+1}\}$.

VIII. Вычислить значение шагового множителя

$$\rho_{k+1} = \min\{\rho_k (1 - \rho_k \| \nabla\psi(x^k, u^k) \|^2), 2^{-1} \| \nabla\psi(x^{k+1}, u^{k+1}) \|^{-2}\}$$

(случай сходимости за конечное число итераций, когда $\| \nabla\psi(x^{k+1}, u^{k+1}) \| = 0$ здесь не рассматривается).

IX. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если выполнены предположения 1, то для алгоритма 1 справедливы следующие утверждения:* (*iii*) — *существует такой номер* k_0, *что*

$$\rho_{k+1} = \rho_k (1 - \rho_k \| \nabla\psi(x^k, u^k) \|^2), \quad k \geqslant k_0;$$

(*iv*) — $\lim_{k\to\infty} \rho_k > 0$; (*v*) — *последовательность* $\{x^k, u^k\}_{k=0}^{\infty}$ *сходится*

к некоторой точке (x^*, u^*) множества седловых точек функции Лагранжа (5.68).

Теорема 1'. *Если выполнены предположения 1 и* (***vi***) *— $z^* = (x^*, u^*)$ — некоторая седловая точка функции Лагранжа;* (***vii***) *— функции f_j, $j = 0, 1, \ldots, m$, в точке x^* трижды дифференцируемы;* (***viii***) *— в точке (x^*, u^*) выполняются достаточные условия максимума второго порядка и условия строгой регулярности, то последовательность $\{x^k, u^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, сходится к точке (x^*, u^*) со скоростью геометрической прогрессии.*

Условия (*viii*) обозначают следующее:

1) оператор $\nabla^2_{ux} \varphi^0 (x^*, u^*)$ имеет ранг l, где $\varphi^0 (x, u) \triangleq f_0 (x) + \sum_{j=1}^{l} u_j f_j (x)$, а l — число такое, что ограничения $f_j (x) \geqslant 0$ при $j = 1, \ldots, l$, активны в точке x^*, а при $j = l + 1, \ldots, m$,— пассивны;

2) $u_j^* > 0$ при $j = 1, 2, \ldots, l$;

3) не существует ненулевого вектора h, удовлетворяющего условиям $\nabla^2_{xx} \varphi^0 (x^*, u^*)h = 0$, $\nabla^2_{ux}\varphi^0 (x^*, u^*) h = 0$ (отметим, что при этих условиях (x^*, u^*) — единственная седловая точка функции φ на $\boldsymbol{R}^n \times \boldsymbol{R}^m$).

2. Метод, использующий штрафные функции экспоненциального типа

Задача 2. Найти $\arg\min_{x \in X} f_0 (x)$, где

$$X \triangleq \{x \mid f_j (x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\};$$

$f_i : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $i = 0, 1, \ldots, m$,— непрерывные функции.

Предположение 2. (*i*) — функции f_i, $i = 0, 1, \ldots, m$ — непрерывно дифференцируемы на $\boldsymbol{R}^n$.

В методе штрафных функций экспоненциального типа строится последовательность $\{(x^k, y^k)\}$, которая сходится при соответствующих предположениях линейно к точке (x^*, y^*), где x^* — решение задачи 2, а $u^* = ((y_1^*)^2, \ldots, (y_m^*)^2)$ — множители Лагранжа задачи 2.

Алгоритм 2

Начало. I. Выбрать начальное приближение $x^0 \in \boldsymbol{R}^n$, числа $y_j^0 > 0$, $j = 1, \ldots, m$, и коэффициент штрафа $\alpha > 0$ такие, чтобы

$$\nabla_x \psi (x^0, y^0, \alpha) = 0 \quad (y^0 = (y_1^0, \ldots, y_m^0)),$$

где экспоненциальная штрафная функция $\psi : \boldsymbol{R}^n \times \boldsymbol{R}^m_+ \times \boldsymbol{R}^1_+ \to \boldsymbol{R}^1$ определяется по формуле

$$\psi (x, y, \alpha) \triangleq f_0 (x) + (1/\alpha) \sum_{j=1}^{m} (y_j)^2 [\exp (\alpha f_j (x)) - 1];$$

$$(y = (y_1, \ldots, y_m)).$$

II. Положить $k = 0$.

Основной цикл. III. Вычислить вектор x^{k+1} так, чтобы

$$\nabla_x \psi(x^{k+1}, y^k, \alpha) = 0. \tag{6.70}$$

Если точка x^{k+1}, удовлетворяющая (6.70), не единственная, то выбрать из них точку x^{k+1}, ближайшую к точке x^k по некоторой норме в $\boldsymbol{R}^n$.

IV. Вычислить вектор $y^{k+1} = (y_1^{k+1}, \ldots, y_m^{k+1})$ по формуле

$$y_j^{k+1} = y_j^k \exp(\alpha f_j(x^{k+1})/2), \quad j = 1, \ldots, m.$$

V. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполняется предположение 2 и (ii) — функции f_i, $i = 0, 1, \ldots, m$ — выпуклые; (iii) — для любых $\mathcal{I}_1, \mathcal{I}_2 \subset \{1, \ldots, m\}$, $\mathcal{I}_1 \neq \mathcal{I}_2$, подзадачи:*

найти $\arg\min_{x \in X_1} f_0(x)$,

$$X_1 \triangleq \{x \mid f_j(x) \leqslant 0, \quad j \in \mathcal{I}_1, \quad x \in \boldsymbol{R}^n\},$$

и найти $\arg\min_{x \in X_2} f_0(x)$,

$$X_2 \triangleq \{x \mid f_j(x) \leqslant 0, \quad j \in \mathcal{I}_2, \quad x \in \boldsymbol{R}^n\},$$

имеют оптимальные решения $x^{1,*}$ и $x^{2,*}$, *соответственно, причем* $f_0(x^{1,*}) \neq f_0(x^{2,*})$.

Тогда любая предельная точка (x^*, y^*) последовательности $\{(x^k, y^k)\}_{k=0}^{\infty}$, порожденной алгоритмом 2, является парой, состоящей из оптимального решения x^* задачи 2 и множителей Куна — Таккера $(y^*)^2$ для задачи 2.

Замечание 2. Теорема 2 остается справедливой в том случае, если заменить условие (*iii*) одним из следующих условий: (*iii′*) — последовательность $\{(x^k, y^k)\}_{k=0}^{\infty}$ сходится к точке (x^*, y^*); (*iii″*) — предельная точка x^* последовательности $\{x^k\}_{k=0}^{\infty}$ является допустимой точкой для задачи 2, т. е.

$$f_j(x^*) \leqslant 0, \quad j = 1, \ldots, m.$$

Теорема 2′. *Пусть выполняется предположение 2. Тогда для последовательности $\{(x^k, y^k)\}_{k=0}^{\infty}$, порожденной алгоритмом 2, справедливы следующие утверждения:*

1) *для всех $i \in \mathcal{K}$ выполняется неравенство*

$$\liminf_{k \to \infty} f_i(x^k) \leqslant 0,$$

где $\mathcal{K} \triangleq \{i \mid \liminf_{k \to \infty} y_i^k = 0, \quad i = 1, \ldots, m\}$;

2) *если последовательность $\{y^k\}_{k=0}^{\infty}$ сходится к точке $\bar{y}$, то любая предельная точка $(\bar{x}, \bar{y})$ последовательности $\{(x^k, y^k)\}_{k=0}^{\infty}$*

такова, что

$$\nabla_x \varphi(\bar{x}, \bar{u}) = 0, \quad \bar{y}_j f_j(\bar{x}) = 0, \quad j = 1, \ldots, m,$$

где

$$\bar{u} = ((\bar{y}_1)^2, \ldots, (\bar{y}_m)^2), \quad \varphi(x, u) \triangleq f_0(x) + \sum_{j=1}^{m} u_j f_j(x), \quad u \in \boldsymbol{R}_+^m;$$

3) *если последовательность* $\{x^k\}_{k=0}^{\infty}$ *сходится к точке* $\bar{x}$, *то любая предельная точка* $(\bar{x}, \bar{y})$ *последовательности* $\{(x^k, y^k)\}_{k=0}^{\infty}$, *такова, что*

$$\nabla_x \varphi(\bar{x}, \bar{u}) = 0 \quad (\bar{u} = ((\bar{y}_1)^2, \ldots, (\bar{y}_m)^2));$$

$$\bar{y}_j f_j(\bar{x}) = 0, \quad f_j(\bar{x}) \leqslant 0, \quad j = 1, \ldots, m.$$

Замечание 2'. Если выполняется предположение 2 и если последовательность $\{(x^k, y^k)\}_{k=0}^{\infty}$, порожденная алгоритмом 2, сходится к точке $(\bar{x}, \bar{y})$, то точка $(\bar{x}, \bar{u})$, где $\bar{u} = ((\bar{y}_1)^2, \ldots, (\bar{y}_m)^2)$ удовлетворяет условиям Куна — Таккера для задачи 2.

Теорема 2''. *Пусть выполняются условия:* (*iv*) — *существует оптимальное локальное решение* x^* *исходной задачи 2;*
(*v*) —

$$\{1, \ldots, l\} = \{j \mid f_j(x^*) = 0, \quad j = 1, \ldots, m\}, \quad l \leqslant m;$$

(*vi*) — *функции* f_i, $i = 0, 1, \ldots, m$ — *дважды непрерывно дифференцируемы в открытой окрестности точки* x^*; (*vii*) — *существует всего лишь единственный вектор множителей Лагранжа* $u^* \in \boldsymbol{R}_+^m$ *такой, что пара* (x^*, u^*) — *есть точка Куна — Таккера, удовлетворяющая достаточным условиям оптимальности второго порядка и* x^* — *регулярная точка для* f_j, $j \in \{1, \ldots, l\}$; (*viii*) — *выполнены условия строгой дополняющей нежесткости в точке* (x^*, u^*), *т. е.* $u_j^* > 0$, *если* $f_j(x^*) = 0$.

Тогда для достаточно большого конечного числа α *существуют окрестности* $V(x^*)$ *и* $W(y^*)$ *точек* x^* *и* y^* *такие, что для любой точки* $y^0 \in W(y^*)$ *существует всего лишь единственная точка* $x^0 \in \bar{V}(x^*)$ (*где* $\bar{V}(x^*)$ — *замыкание множества* $V(x^*)$) *такая, что* $\nabla_x \psi(x^0, y^0, \alpha) = 0$ *и последовательность* $\{(x^k, y^k)\}_{k=0}^{\infty}$ *сходится к точке* (x^*, y^*) *линейно. Более того, если* α *стремится к бесконечности, то последовательность* $\{y^k\}_{k=0}^{\infty}$ *сходится к* y^* *сверхлинейно.*

Библиографические указания. При написании пункта 1 использована работа [233], пункт 2 основан на результатах работы [532].

В работе [550] исследуются свойства расширенных лагранжианов.

5.16. Методы нагруженного функционала

В приводимых в этом параграфе методах на k-й итерации требуется искать абсолютный минимум по x нагруженного функционала

$$\psi(x,\ \alpha_k) \triangleq [f_0(x) - \alpha_k]^2 + \sum_{i=1}^{m} (f_i(x))^2,$$

где числовая последовательность $\{\alpha_k\}_{k=0}^{\infty}$ строится в алгоритмах таким образом, чтобы обеспечить равенство

$$\lim_{k\to\infty} \alpha_k = \min_{x\in X} f_0(x).$$

1. Общий случай

Задача 1. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного множества

$$X \triangleq \{x \mid f_i(x) = 0,\quad i = 1,\ \ldots,\ m,\quad x \in \boldsymbol{R}^n\},$$

где $f_i : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $m < n$.

В алгоритме 1 на k-й итерации вычисляют точку

$$x^k = \arg\min_{x\in \boldsymbol{R}^n} \psi(x,\ \alpha_k),$$

$$\psi(x,\ \alpha_k) \triangleq [f_0(x) - \alpha_k]^2 + \sum_{i=1}^{m} (f_i(x))^2,$$

где $\alpha_k \to \min_{x\in X} f_0(x)$ при $k \to \infty$.

Предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$ являются решениями задачи 1. Кроме того, на каждой итерации вычисляется отрезок $[\alpha^-,\ \alpha^+]$, содержащий значение $\min_{x\in X} f_0(x)$, и длина которого при $k \to \infty$ стремится к нулю.

Алгоритм 1

Начало I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Положить $\alpha_0 = f_0(x^0)$.

III. Положить $k = 1$.

IV. Определить функцию $\psi(x,\ \alpha) : \boldsymbol{R}^n \times \boldsymbol{R}^1 \to \boldsymbol{R}^1$ по правилу

$$\psi(x,\ \alpha) = (f_0(x) - \alpha)^2 + \sum_{i=1}^{m} (f_i(x))^2.$$

V. Выбрать константу $\delta_0 > 0$ (δ_0 — достаточно малая величина) и положить $\delta = \delta_0$.

Основной цикл. VI. Вычислить

$$\alpha_k = \alpha_{k-1} - \delta.$$

VII. Вычислить вектор x^k, удовлетворяющий условию

$$\psi(x^k, \alpha_k) = \min_{x \in R^n} \psi(x, \alpha_k).$$

VIII. Если выполняется неравенство

$$[\psi(x^k, \alpha_k)]^{1/2} \leqslant \text{eps (eps — «машинный ноль»)},$$

то положить $\delta = 2\delta$, $k = k + 1$ и перейти к шагу VI; иначе положить $\alpha^+ = \alpha_{k-1}$ и перейти к шагу IX.

IX. Положить $\alpha^- = \alpha_k$ и перейти к шагу X.

X. Вычислить значения σ_{k+1} и τ_{k+1}, соответственно, по формулам

$$\sigma_{k+1} = \alpha_k + (\psi(x^k, \alpha_k))^{1/2};$$

$$\tau_{k+1} = \alpha_k + \psi(x^k, \alpha_k) / \left[\psi(x^k, \alpha_k) - \sum_{i=1}^{m} (f_i(x^k))^2 \right]^{1/2}.$$

XI. Положить $k = k + 1$.

XII. Положить $\alpha^- = \sigma_k$.

XIII. Если $\tau_k < \alpha^+$, тогда положить $\alpha_k = \tau_k$ и перейти к шагу XIV; иначе положить $\alpha_k = \sigma_k$ и перейти к шагу XIV.

XIV. Вычислить вектор x^k, удовлетворяющий условию

$$\psi(x^k, \alpha_k) = \min_{x \in R^n} \psi(x, \alpha_k).$$

XV. Если $[\psi(x^k, \alpha_k)]^{1/2} \geqslant$ eps, то перейти к шагу X; иначе положить $\alpha^+ = \alpha_k$ и перейти к шагу XVI.

XVI. Если $\alpha^+ - \alpha^- <$ eps, тогда прекратить вычисления (в этом случае находят решение x^k задачи 1); иначе положить $\alpha_k = \alpha^-$ и перейти к шагу XIV.

Теорема 1. *Если (i) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — непрерывны в любой замкнутой ограниченной области Y пространства R^n; (ii) — задача 1 имеет решение x^*;* eps = 0; *(iii) — $\psi(x^k, \alpha_k) = \min_{x \in R^n} \psi(x, \alpha_k)$ может быть найдено алгоритмами безусловной оптимизации; (iv) — $\psi(x^k, \alpha_k) > 0$ эквивалентно $f_0(x^k) < f_0(x^*)$, то алгоритм 1 сходится и предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$ являются решениями задачи 1.*

2. Выпуклый случай

Задача 2. Найти $\arg\max_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества X, задаваемого соотношением

$$X \triangleq \{x \mid f_i(x) \leqslant 0, \quad i = 1, \ldots, m, \quad x \in R^n\}.$$

Предположения 2. (*i*) — функция $f_0(x)$ — вогнута, функции $f_j(x)$, $j = 1, \ldots, m$ — выпуклы в R^n; (*ii*) — функции $f_j(x)$, $j = 0$,

1, ..., m — дважды непрерывно дифференцируемы в R^n; (iii) — множество X ограничено и непусто.

Метод нагруженного функционала в выпуклом случае при определенных условиях сходится к решению задачи 2 со сверхлинейной скоростью. На каждой итерации алгоритма требуется решать (точно) задачу безусловной минимизации нагруженного функционала. Для начала работы алгоритма необходимо иметь оценку сверху для величины $\max_{x \in X} f_0(x)$.

Алгоритм 2

Начало. I. Найти число α_1, удовлетворяющее условию

$$\alpha_1 \geqslant f_0(x^*),$$

где x^*— решение задачи 2.

II. Определить функцию $\psi(x, \alpha)$ (нагруженный функционал) по правилу

$$\psi(x, \alpha) \triangleq (\max\{0, \alpha - f_0(x)\})^2 + \sum_{i=1}^{m} (\max\{0, f_i(x)\})^2.$$

III. Положить $k = 1$.

Основной цикл. IV. Вычислить вектор x^k, удовлетворяющий условию

$$\psi(x^k, \alpha_k) = \min_{x \in R^n} \psi(x, \alpha_k). \qquad (5.71)$$

V. Если $\psi(x^k, \alpha_k) = 0$, то прекратить вычисления, иначе перейти к шагу VI.

VI. Вычислить

$$\alpha_{k+1} = \alpha_k - \psi(x^k, \alpha_k)(\alpha_k - f_0(x^k))^{-1}.$$

VII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Пусть выполняются предположения 2 и существует решение x^* задачи 2 такое, что имеет решение и задача:*

$$\text{найти } \arg\max_{x \in Y} (\nabla f_0(x^*),\ x - x^*), \qquad (5.72)$$

$$Y = \{x \mid (\nabla f_i(x^*),\ x - x^*) \leqslant 0,\ i \in \{j \mid f_j(x^*) = 0\},\ x \in R^n\}.$$

Тогда для бесконечной последовательности $\{\alpha_k\}_{k=1}^{\infty}$, порожденной алгоритмом 2, существует константа β такая, что $\lim_{k\to\infty} \alpha_k = f_0(x^*)$ и при достаточно больших значениях k выполняется неравенство

$$0 < \alpha_{k+1} - f_0(x^*) \leqslant \beta(\alpha_k - f_0(x^*))^{3/2}.$$

Теорема 2′. *Пусть выполняются предположения 2 и (iv) — существует решение x^* задачи 2 такое, что векторы $\nabla f_i(x^*)$, $i \in \{j \mid f_j(x^*) = 0,\ j = 1, \ldots, m\}$ — линейно независимы; (v) — су-*

ществует число $\delta > 0$, *для которого*

$$\left(y^T \frac{\partial^2 \varphi(p^*, x^*)}{\partial x^2}, y\right) \leqslant -\delta \|y\|^2, \quad \forall y \in R^n,$$

где

$$\varphi(p, x) = f_0(x) - \sum_{i=1}^{m} p_i f_i(x), \quad p = (p_1, \ldots, p_m);$$

$$(\nabla f(x^*))^T p^* = \nabla f_0(x^*), \quad f(x) = (f_1(x), \ldots, f_m(x));$$

$$(p^*, f(x^*)) = 0, \quad p^* \geqslant 0.$$

Тогда для бесконечной последовательности $\{x^k\}_{k=1}^{\infty}$, *порожденной алгоритмом 2, существует константа* $\eta \geqslant 0$ *такая, что при достаточно больших* k *выполняется неравенство*

$$\|x^k - x^*\| \leqslant \eta(\alpha_k - f_0(x^*))^{1/4} \|x^{k-1} - x^*\|,$$

т. е. последовательность $\{x^k\}_{k=1}^{\infty}$ *сходится к* x^* *сверхлинейно.*

3. Приближенная схема

В приводимом ниже алгоритме при известной оценке сверху значения $\max\limits_{x \in X} f_0(x)$ за конечное число итераций вычисляется приближенное решение $\bar{x}$ задачи 2 такое, что

$$(f_0(\bar{x}) - f_0(x^*))^2 - \sum_{i=1}^{m} (\max\{0, f_i(\bar{x})\})^2 \leqslant \varepsilon_2,$$

где x^* — решение задачи 2; ε_2 — наперед заданная константа. При этом на каждой итерации алгоритма задачу безусловной минимизации нагруженного функционала достаточно решать также приближенно.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать произвольное число α_1 (обычно $\alpha_1 \geqslant f_0(x^*)$, где x^* — решение задачи 2).

III. Задать произвольные константы $\varepsilon_1 > 0$ и $\varepsilon_2 > 0$, определяющие точности решения вспомогательной задачи и задачи 2 (рекомендуется $\varepsilon_1 \in [10^{-5}, 10^{-1}]$, $\varepsilon_2 \in [10^{-5}, 10^{-1}]$).

IV. Определить нагруженный функционал

$$\psi(x, \alpha) \triangleq (\max\{0, \alpha - f_0(x)\})^2 + \sum_{i=1}^{m} (\max\{0, f_i(x)\})^2.$$

V. Положить $k = 1$.

Основной цикл. VI. Вычислить вектор x^k, для которого выполняется неравенство

$$\psi(x^k, \alpha_k) \leqslant \psi(x^{k-1}, \alpha_k), \tag{5.73}$$

а также либо неравенства

$$\|\nabla_x \psi(x^k, \alpha_k)\| < \varepsilon_1 (\alpha_k - f_0(x^k)), \quad \psi(x^k, \alpha_k) > \varepsilon_2, \tag{5.74}$$

либо

$$\psi(x^k, \alpha_k) \leqslant \varepsilon_2. \tag{5.75}$$

VII. Если выполняются неравенства (5.73) и (5.75), то прекратить вычисления; если (5.73) и (5.74), то перейти к шагу VIII.

VIII. Вычислить

$$\alpha_{k+1} = \alpha_k - \psi(x^k, \alpha_k)(\alpha_k - f_0(x^k))^{-1}. \tag{5.76}$$

IX. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 3. *Пусть выполняются условия (i), (iii) предположения 2 и функции $f_j(x)$, $j = 0, 1, \ldots, m$ — непрерывно дифференцируемы. Тогда для произвольного начального приближения $x^0 \in \boldsymbol{R}^n$ и любых значений α_1, $\varepsilon_1 > 0$, $\varepsilon_2 > 0$ алгоритм 3 остановится за конечное число шагов, причем точки x^k его остановки, полученные при фиксированных x^0, $\alpha_1 \geqslant f_0(x^*)$ и различных ε_1, ε_2, будут при $\varepsilon_1 \to 0$, $\varepsilon_2 \to \to 0$ сходиться к множеству решений задачи 2.*

Теорема 3'. *Пусть выполняются предположения теоремы 3 и, кроме того: (i) — функция $f_0(x)$ — дважды непрерывно дифференцируема; (ii) — существует константа μ такая, что для любых $x, y \in \boldsymbol{R}^n$ выполняется неравенство*

$$\left(y^T \frac{\partial^2 f_0(x)}{\partial x^2}, y\right) \leqslant -\mu \|y\|^2;$$

(iii) — производная $\partial f_0(x^*)/\partial x$ не равна нулю; (iv) — в алгоритме 3 неравенства (5.74) заменены следующими:

$$\alpha_k > f_0(x^k);$$

$$\|\nabla_x \psi(x^k, \alpha_k)\| \leqslant \varepsilon_1 \min \left\{ \left(\sum_{i=1}^{m} (\max\{0, f_i(x^k)\})\right)^2, \right.$$

$$\left. \left[\sum_{i=1}^{m} (\max\{0, f_i(x^k)\})^2 (\alpha_k - f_0(x^k))\right]^{1/2} \right\}; \tag{5.77}$$

$$\psi(x^k, \alpha_k) > \varepsilon_2;$$

(v) — функция $f_0(x)$ строго вогнута.

Тогда при достаточно малых $\varepsilon_1 > 0$ и любых x^k, α_k, $\varepsilon_2 > 0$, удовлетворяющих условиям (5.77), значение α_{k+1}, вычисленное на шаге VIII алгоритма 3 по (5.76), будет не меньше, чем $f_0(x^*)$.

Библиографические указания. Пункт 1 написан на основании работы [509], при написании пункта 2 использовались работы [216, 217, 218], пункт 3 основан на результатах работы [217]. Дополнительные сведения о методах нагруженного функционала и их вариантах, примеры практического использования методов можно найти в работах [302, 322, 508].

5.17. Методы штрафных оценок

Задача 0. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0: R^n \to R^1$ и заданного множества

$$X \triangleq \{x \mid f_j(x) = 0, \quad j = 1, \ldots, m, \quad x \in R^n\}.$$

Метод штрафных оценок является комбинацией методов штрафных функций и множителей Лагранжа и основан на построении модифицированной функции Лагранжа ψ: $R^n \times R^m \times R^1_+ \to R^1$:

$$\psi(x, y, \alpha) \triangleq f_0(x) + \sum_{j=1}^{m} y_j f_j(x) + \frac{\alpha}{2} \sum_{j=1}^{m} f_j^2(x), \qquad (5.78)$$

где $\alpha \geqslant 0$, $y = (y_1, \ldots, y_m) \in R^m$.

В методе ведутся итерации как по исходным переменным x, так и по двойственным переменным y. В детерминированном случае на каждой итерации решается вспомогательная задача на безусловный минимум. Метод штрафных оценок в детерминированном случае локально сходится со скоростью геометрической прогрессии, знаменатель которой тем меньше, чем больше коэффициент штрафа α. При наличии случайных помех метод штрафных оценок сходится к локальному решению с вероятностью $1 - \delta$ (ибо без предположения выпуклости функций f_j сходимости с вероятностью 1 не существует).

1. Детерминированный случай

Алгоритм 1

Начало. I. Выбрать начальное приближение $y^0 \in R^m$ двойственной переменной.

II. Положить $k = 0$.

Основной цикл. III. Выбрать коэффициент штрафа α_k.

IV. Вычислить следующее приближение x^{k+1} как точку безусловного минимума функции $\psi(x, y^k, \alpha_k)$ (где $\psi(x, y, \alpha)$ определяется по (5.78)), т. е.

$$\psi(x^{k+1}, y^k, \alpha_k) = \min_x \psi(x, y^k, \alpha_k).$$

V. Вычислить следующее приближение y^{k+1} двойственной переменной по формуле

$$y^{k+1} = y^k + \alpha_k f(x^{k+1}),$$

где

$$f(x^{k+1}) = (f_1(x^{k+1}), \ldots, f_m(x^{k+1})).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются условия: (i) — существует локальная точка минимума x^*, т. е.*

$$f_j(x^*) = 0, \quad j = 1, \ldots, m, \quad и \quad f_0(x^*) \leqslant f_0(x)$$

при

$$f_j(x) = 0, \quad j = 1, \ldots, m, \quad \|x - x^*\| \leqslant \varepsilon, \quad \varepsilon > 0;$$

(*ii*) — *в ε-окрестности точки x^* существуют первые и вторые производные функций $f_j(x)$, $j = 0, 1, \ldots, m$, причем $\nabla^2_{xx} f_j(x)$, $j = 0, 1, \ldots, m$, удовлетворяют условиям Липшица*

$$\|\nabla^2_{xx} f_j(x) - \nabla^2_{xx} f_j(\bar{x})\| \leqslant \gamma_1 \|x - \bar{x}\|, \quad \gamma_1 < \infty;$$

(*iii*) — *градиенты $\nabla f_j(x^*)$, $j = 1, \ldots, m$ — линейно независимы;*
(*iv*) — *выполняется достаточное условие минимума*

$$(\nabla^2_{xx}\varphi(x^*, y^*)x, x) \geqslant \delta_1 \|x\|^2, \ \delta_1 > 0 \ \text{при} \ (\nabla^2_{yx}\varphi(x^*, y^*))x = 0.$$

Тогда для всякого $\rho > 0$ найдется число $\beta = \beta(\rho)$ такое, что при $\alpha_k \geqslant \beta$, $k = 0, 1, \ldots$, алгоритм 1 (в котором под x^{k+1} понимается точка локального минимума функции $\psi(x, y^k, \alpha_k)$ из окрестности x^*) сходится при любом начальном приближении $y^0 \in S_\rho$, где

$$S_\rho \triangleq \{y \mid \|y - y^*\| \leqslant \rho, \quad y \in \boldsymbol{R}^m\},$$

причем для всех $k \geqslant 1$ справедливы оценки

$$\|x^k - x^*\| \leqslant (\theta_0)^k \|y^0 - y^*\| / (\alpha_0 \alpha_1 \ldots \alpha_{k-1});$$

$$\|y^k - y^*\| \leqslant (\theta_0)^k \|y^0 - y^*\| / (\alpha_0 \alpha_1 \ldots \alpha_{k-1}),$$

где $\theta_0 > 0$ — некоторая константа.

При условиях (*i*) — (*iii*) применимо правило множителей Лагранжа, т. е. найдется $y^* \in \boldsymbol{R}^m$ такой, что $\nabla_x\varphi(x^*, y^*) = 0$, где $\nabla_x\varphi(x^*, y^*)$ — производная функции Лагранжа

$$\varphi(x, y) \triangleq f_0(x) + \sum_{j=1}^{m} f_j(x) y_j.$$

Условия теоремы (условия минимума) выражают условия невырожденности локального минимума и являются минимально ограничительными.

Замечание 1. Из теоремы 1 следует, что при $\alpha_k = \bar{\alpha}$ для достаточно больших $\bar{\alpha}$ алгоритм 1 сходится не медленнее геометрической прогрессии со знаменателем $q = \theta_0/\bar{\alpha}$. Причем за счет увеличения $\bar{\alpha}$ этот знаменатель может быть сделан сколько угодно малым.

Замечание 1'. Алгоритм 1 можно модифицировать таким образом, чтобы на шаге IV лишь приближенно минимизировать функцию $\psi(x, y^k, \alpha_k)$. Именно в качестве x^{k+1} выбирать любую точку из окрестности x^*, в которой

$$\|\nabla_x\psi(x^{k+1}, y^k, \alpha_k)\| \leqslant \mu \|f(x^{k+1})\|,$$

где $\mu \geqslant 0$ — наперед заданная константа.

В условиях теоремы 1 для модифицированного алгоритма справедливы оценки

$$\|x^k - x^*\| \leqslant (\theta_0(1+\mu))^k \|y^0 - y^*\|/(\alpha_0\alpha_1 \ldots \alpha_{k-1});$$
$$\|y^k - y^*\| \leqslant (\theta_0(1+\mu))^k \|y^0 - y^*\|/(\alpha_0\alpha_1 \ldots \alpha_{k-1}),$$

(константа θ_0 та же, что и в теореме 1).

Замечание 1''. Если известно приближение x^0 для x^*, то начальное приближение y^0 в алгоритме 1 можно найти из условия минимума квадратичного функционала

$$q_0(y) \triangleq \|\nabla f_0(x^0) + (\nabla^2_{yx}\varphi(x^0, 0))^T y\|^2.$$

Чем ближе x^0 к x^*, тем ближе будет найденное y^0 к y^*.

Замечание 1'''. Наиболее существенной трудностью в применении алгоритма 1 является проблема локальных минимумов функции $\psi(x, y^k, \alpha_k)$: неясно, как выбрать из них тот, который находится в окрестности x^*.

2. Стохастический случай

Алгоритм 2

Начало. I. Выбрать произвольные начальные приближения $x^0 \in R^n$ и $y^0 \in R^m$, основной и двойственной переменных.

II. Выбрать коэффициент штрафа $\alpha_0 > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Найти шаговый множитель ρ_k, удовлетворяющий условиям теоремы 2.

V. Вычислить реализацию ξ^k случайного вектора $\bar{\xi}^k$, удовлетворяющего условию

$$E(\bar{\xi}^k/x^0, y^0, \ldots, x^k, y^k) = \nabla f_0(x^k).$$

VI. Вычислить реализацию ζ^k случайного вектора $\bar{\zeta}^k$, удовлетворяющего условию

$$E(\bar{\zeta}/x^0, y^0, \ldots, x^k, y^k) = f(x^k),$$

где $f(x^k)$ — вектор-столбец с компонентами $f_1(x^k)$, $f_2(x^k)$, ..., $f_m(x^k)$.

VII. Вычислить матрицу A_k размера $m \times n$, строками которой являются реализации $(v_i^k)^T$ случайных векторов $(\bar{v}_i^k)^T$, $i = 1, \ldots$ ..., m, удовлетворяющих условиям

$$E(\bar{v}_i^k/x^0, y^0, \ldots, x^k, y^k) = \nabla f_i(x^k), \quad i = 1, \ldots, m.$$

VIII. Вычислить следующее приближение x^{k+1} для основной переменной по формуле

$$x^{k+1} = x^k - \rho_k(\xi^k + (A_k)^T y^k + \alpha_0 (A_k)^T \zeta^k).$$

IX. Вычислить следующее приближение y^{k+1} для двойственной переменной по формуле

$$y^{k+1} = y^k + \rho_k \zeta^k.$$

X. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. Пусть выполняются условия (*i*) — (*iv*) *теоремы 1 и* (*v*) — *случайные вектора* $\bar{\xi}^k$, $\bar{\zeta}^k$, $\bar{\nu}_i^k$, $i = 1, \ldots, m$ — *взаимно независимы при различных* k; (*vi*) — *случайные вектора* $\bar{\zeta}^k$ и $\bar{\nu}_i^k$, $i = 1, \ldots, m$ — *взаимно независимы;* (*vii*) — *сумма дисперсий компонент каждого из векторов* $\bar{\xi}^k$, $\bar{\zeta}^k$, $\bar{\nu}_i^k$, $i = 1, \ldots, m$ — *ограничена числом* σ^2; (*viii*) — *шаговые множители* ρ_k *таковы, что*

$$\rho_k \geqslant 0, \quad k = 0, 1, \ldots; \quad \sum_{k=0}^{\infty} \rho_k = \infty; \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty;$$

(*ix*) —

$$E(\|x^0\|^2 + \|y^0\|^2) < \infty.$$

Тогда найдется такое число $\bar{\alpha} > 0$, *что при* $\alpha_0 \geqslant \bar{\alpha}$ *последовательность* $\{(x^k, y^k)\}_{k=0}^{\infty}$, *порожденная алгоритмом 2, удовлетворяет условию*

$$P\{x^k \to x^*, \ y^k \to y^*\} \geqslant 1 - \delta,$$

где

$$\delta = \gamma_2 E(\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2) + \gamma_3 \sigma^2 \sum_{k=0}^{\infty} \rho_k^2;$$

γ_2, γ_3 — *константы, зависящие от детерминированной части задачи 1.*

При этом, если $k\rho_k \to \rho$, $k\rho_k$ — *монотонно возрастает, то для всех* k *при* $\gamma_4 > 0$ *выполняется*

$$P\{\|x^k - x^*\|^2 + \|y^k - y^*\|^2 \leqslant \gamma_4 \rho_k\} \geqslant 1 - \delta - 1/\gamma_4.$$

В частности, по всякому $\delta_0 > 0$ *можно выбрать* $\bar{\beta}$, ρ, $\bar{\mu}$, γ_4 *так, что при*

$$\rho_k = \rho/(k + \bar{\mu}), \quad E(\|x^0 - x^*\|^2 + \|y^0 - y^*\|^2) \leqslant \bar{\beta}$$

для алгоритма 2 справедливо

$$P\{\|x^k - x^*\|^2 + \|y^k - y^*\|^2 \leqslant \gamma_4/(k + \bar{\mu})\} \geqslant 1 - \delta_0.$$

3. Метод штрафных оценок для задач выпуклого программирования

З а д а ч а 3. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in Q\},$$

где Q — множество в R^n; $f_j : R^n \to R^1$, $j = 1, \ldots, m$ — заданные функции.

Предположения 3. (*i*) — множество Q — выпукло и замкнуто; (*ii*) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — выпуклые и непрерывные.

В выпуклом случае при естественных предположениях метод штрафных оценок сходится к множеству решений прямой и двойственной задач. При выполнении условий регулярности решения задачи 3 метод локально сходится со скоростью геометрической прогрессии, причем за счет увеличения штрафного коэффициента знаменатель геометрической прогрессии может быть сделан сколь угодно малым.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение двойственной переменной $y^0 \in \boldsymbol{R}_+^m$.

II. Определить модифицированную функцию Лагранжа $\psi(x, y, \alpha) : \boldsymbol{R}^n \times \boldsymbol{R}_+^m \times \boldsymbol{R}_+^1 \to \boldsymbol{R}^1$,

$$\psi(x, y, \alpha) \triangleq f_0(x) + \frac{1}{2\alpha} \sum_{i=1}^{m} ([y_i + \alpha f_i(x)]_+)^2 - \frac{1}{2\alpha} \sum_{i=1}^{m} (y_i)^2.$$

(Здесь и далее $[t]_+ \triangleq \max\{0, t\}$).

III. Положить $k = 0$.

Основной цикл. IV. Найти штрафной коэффициент $\alpha_k > 0$, удовлетворяющий условию $\alpha_k \geqslant \alpha > 0$, где $\alpha > 0$ — произвольная константа.

V. Вычислить точку $x^k \in Q$, удовлетворяющую условию

$$\psi(x^k, y^k, \alpha_k) = \min_{x \in Q} \psi(x, y^k, \alpha_k).$$

VI. Положить $h^k = (f_1(x^k), \ldots, f_m(x^k))$.

VII. Вычислить точку

$$y^{k+1} = [y^k + \alpha_k h^k]_+.$$

VIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 3. *Пусть выполняются предположения 3 и пусть* (*iii*) — *множество решений* X^* *задачи 3 непусто и ограничено;* (*iv*) — *множество решений* Y^*, *двойственной к задаче 3, непусто.*

Тогда алгоритм 3 при любом $y^0 \geqslant 0$ (*и любом* $\alpha > 0$) *порождает ограниченные последовательности* $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$ *такие, что*

$$\min_{x^* \in X^*} \|x^k - x^*\| \to 0 \quad \text{при} \quad k \to \infty;$$

$$\min_{y^* \in Y^*} \|y^k - y^*\| \to 0 \quad \text{при} \quad k \to \infty;$$

$$\psi(x^k, y^k, \alpha_k) \leqslant f_0(x^*), \quad x^* \in X^*.$$

Двойственной к задаче 3 является следующая задача:

$$\text{найти } \arg\max_{y \geqslant 0} g_0(y),$$

где $g_0(y) \triangleq \inf_{x \in Q} \varphi(x, y)$, а $\varphi(x, y)$ — функция Лагранжа задачи 3, т. е.

$$\varphi(x, y) \triangleq f_0(x) + \sum_{i=1}^{m} y_i f_i(x), \quad y = (y_1, \ldots, y_m) \in \boldsymbol{R}_+^m, \quad x \in Q.$$

Множество $X^* \times Y^*$ является множеством седловых точек функции $\varphi(x, y)$.

Теорема 3'. *Пусть выполнены предположения 3 и пусть: (iv) — в задаче 3 множество $Q = \boldsymbol{R}^n$; (v) — задача 3 имеет решение x^*; (vi) — в некоторой окрестности x^* функции $f_j(x)$, $j = 0, 1, \ldots, m$ — дважды непрерывно дифференцируемы; (vii) —* векторы $\nabla f_i(x^*)$ *при $i \in \mathcal{I}^* \triangleq \{i \mid f_i(x^*) = 0,\ i \in [1:m]\}$ — линейно независимы; (viii) — если $(\nabla f_i(x^*), x) = 0$ для всех $i \in \mathcal{I}^*$, то*

$$(\nabla_{xx}^2 \varphi(x^*, y^*) x, x) \geqslant \beta_1 \|x\|^2, \quad \beta_1 > 0,$$

где $\nabla_{xx}^2 \varphi(x^*, y^*)$ — матрица вторых частных производных функции $\varphi(x, y^*)$ в точке x^*; $\varphi(x, y)$ — функция Лагранжа задачи 3; y^* — множители Лагранжа задачи 3; (ix) — выполнены условия строгой дополняющей нежесткости, т. е. $y_i^* > 0$ при $i \in \mathcal{I}^*$.

Тогда для любого $\alpha > 0$ найдется $\delta > 0$ такое, что при $\|y^0 - y^*\| < \delta$ последовательности $\{x^k\}_{k=0}^{\infty}$, $\{y_k\}_{k=0}^{\infty}$, порожденные алгоритмом 3, удовлетворяют при всех $k \geqslant 0$ неравенствам

$$\|x^k - x^*\| \leqslant \gamma_1 \|y^k - y^*\|, \quad \gamma_1 > 0; \tag{5.79}$$
$$\|y^{k+1} - y^*\| \leqslant q_1 \|y^k - y^*\|, \quad 0 < q_1 < 1,$$

т. е. последовательности $\{x^k\}_{k=0}^{\infty}$ и *$\{y^k\}_{k=0}^{\infty}$ сходятся со скоростью геометрической прогрессии, соответственно, к x^* и y^*.*

Следствие. При условиях теоремы 3' для любых $\alpha > 0$, $y^0 \geqslant 0$ найдется такое k_0, что при $k \geqslant k_0$ последовательности $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$, порожденные алгоритмом 3, удовлетворяют неравенствам (5.79).

Теорема 3". *Пусть выполняются все предположения теоремы 3' и, кроме того, (x) — матрицы $\nabla_{xx}^2 f_0(x)$ и $\nabla_{xx}^2 f_i(x)$, $i \in \mathcal{I}^*$, удовлетворяют условию Липшица в некоторой окрестности точки x^*.*

Тогда для любого $y^0 \geqslant 0$ найдется столь большое число $\alpha = \alpha(\|y^0 - y^*\|) > 0$, что при $\alpha_k \geqslant \alpha$ последовательности $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$, порожденные алгоритмом 3, для всех $k \geqslant 0$ удовлетворяют неравенствам

$$\|x^k - x^*\| \leqslant \frac{\gamma_2}{\alpha_k} \|y^k - y^*\|;$$

$$\|y^{k+1} - y^*\| \leqslant \frac{\gamma_2}{\alpha_k} \|y^k - y^*\|,$$

где постоянная $\gamma_2 > 0$ не зависит от α_k.

Библиографические указания. Пункт 1 написан на основании работы [302], пункт 2 основан на результатах работы [300], пункт 3 — на результатах работы [357]. Рекомендации к подбору начальных значений y^0 и α_0 в алгоритме 1 можно найти в работах [293, 487, 493, 541], частные варианты алгоритма 2 приведены в [411].

В [330] для невыпуклой экстремальной задачи с ограничениями в форме неравенств доказана локальная сходимость метода штрафных оценок со скоростью геометрической прогрессии. В [302] изучался метод штрафных оценок для невыпуклых задач с ограничениями в форме равенств, в [301] — для задач линейного программирования.

В работе [432] для решения задачи оптимизации с ограничениями типа равенств используется модифицированная функция Лагранжа и алгоритм типа штрафных оценок.

5.18. Методы проектирования обобщенного градиента

1. Основной алгоритм

Задача 1. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного множества $X \subset \boldsymbol{R}^n$.

Предположения 1. (*i*) — функция f_0 — выпукла вниз; (*ii*) — множество X выпукло и замкнуто.

Этот метод напоминает метод проекции градиентов, если вместо градиента использовать обобщенный градиент $\hat{\nabla} f_0$ функции f_0.

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in \boldsymbol{R}^n$ и положить $k = 0$.

Основной цикл. II. Вычислить обобщенный градиент $\hat{\nabla} f_0(x^k)$ функции f_0 в точке x^k.

III. Вычислить значения шагового множителя ρ_k и нормирующего множителя γ_k, удовлетворяющие условиям теоремы 1.

IV. Вычислить следующее приближение

$$x^{k+1} = \pi_X(x^k - \rho_k\gamma_k\hat{\nabla} f_0(x^k)).$$

V. Положить $k = k + 1$ и перейти к шагу II.

Теорема 1. *Пусть выполнены предположения 1 и условия:* (*i*) — *существует точка x^*, принадлежащая множеству минимумов X^* функции f_0 в области X, для которой $\| x^* \| \leqslant \text{const}$;* (*ii*) — *для любого числа $\alpha < \infty$ найдется такое число $\beta(\alpha) < \infty$, что $\| \hat{\nabla} f_0(x) \| \leqslant \beta(\alpha)$ при $\| x \| \leqslant \alpha$;* (*iii*) — *нормирующие множители γ_k, $k = 0, 1, \ldots$, таковы, что $\gamma_k > 0$ и $\gamma_k \| \hat{\nabla} f_0(x^k) \| \leqslant \text{const}$, $k = 0, 1, \ldots$,*

где $\{x^k\}_{k=0}^{\infty}$ — последовательность, порожденная алгоритмом 1;

(*iv*) — *шаговые множители ρ_k, $k = 0, 1, \ldots$, удовлетворяют условиям*

$$\rho_k \geqslant 0, \quad \rho_k \to 0, \quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

Тогда найдется такая подпоследовательность $\{f_0(x^{k_s})\}_{s=0}^{\infty}$ *последовательности* $\{f_0(x^k)\}_{k=0}^{\infty}$, *что*

$$\lim_{s\to\infty} f_0(x^{k_s}) = f_0(x^*),$$

т. е.

$$\lim_{k\to\infty} \min_{s\leqslant k} f_0(x^s) = f_0(x^*).$$

Теорема 1'. *Если в дополнение к условиям теоремы 1 множество минимумов* X^* *функции* f_0 *в области* X *ограничено, то последовательность* $\{f_0(x^k)\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, сходится к* $f_0(x^*)$ *и*

$$\inf_{x^*\in X^*} \|x^* - x^k\| \to 0 \quad \text{при} \quad k\to\infty.$$

Замечание 1. Если в теореме 1 потребовать, чтобы вместо (*iv*) выполнялись условия

$$\rho_k \geqslant 0, \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, будет сходиться к решению $x^* \in X^*$ без предположения об ограниченности множества X^*.

2. Многошаговый метод обобщенного градиентного спуска

Задача 2. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X \triangleq \{x \mid f_1(x) \leqslant 0, \quad x \in R^n\}.$$

Предположения 2. (*i*) — функции f_0 и f_1 — выпуклые на R^n; (*ii*) — существует точка $\bar{x} \in R^n$ такая, что $f_1(\bar{x}) < 0$; (*iii*) — множество X — ограничено.

В приводимом ниже алгоритме направление спуска выбирается с использованием обобщенных градиентов и значений функций f_0 и f_1 на предыдущих итерациях. Шаговые множители ρ_k удовлетворяют классическим условиям.

Алгоритм 2

Начало. I. Выбрать произвольное натуральное число $m \geqslant 1$.

II. Выбрать произвольный набор точек $\{x^{-m+1}, \ldots, x^0\}$.

III. Выбрать константу α, удовлетворяющую условию $\alpha > f_0(x^*)$, где x^* — решение задачи 2.

IV. Выбрать константы $\delta_1 > 0$, $\delta_2 > 0$, причем δ_2 выбрать из условия $\delta_2 < -f_1(\bar{x})$ (точка $\bar{x}$ определяется в условии (*ii*) предположения 2).

V. Положить $k = 0$.

Основной цикл. VI. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям теоремы 2.

VII. Если $f_0(x^k) > \alpha$, то положить $\mathcal{I}_k = \{k\}$ и перейти к шагу VIII; если $f_0(x^k) \leqslant \alpha$, то положить $\mathcal{I}_k = \{k-m+1, \ldots, k\}$ и перейти к шагу VIII.

VIII. Вычислить $\hat{\nabla} f_0(x^k)$ и $\hat{\nabla} f_1(x^k)$ — обобщенные градиенты функций f_0 и f_1 в точке x^k.

IX. Вычислить вектор h^k из условия

$$\min_{\|h\| \leqslant \rho_k} \varphi_k(h) = \varphi_k(h^k),$$

где функция φ_k определяется по правилу

$$\varphi_k(h) \triangleq \max_{j \in \mathcal{I}_k} \psi_{kj}(h),$$

$$\psi_{kj}(h) = \begin{cases} f_0(x^j) - f_0(x^k) + (x^k - x^j,\ \hat{\nabla} f_0(x^j)) + (h,\ \hat{\nabla} f_0(x^j)), \\ \qquad\qquad f_1(x^j) < -\delta_2, \\ \max\{f_0(x^j) - f_0(x^k) + (x^k - x^j,\ \hat{\nabla} f_0(x^j)) + (h,\ \hat{\nabla} f_0(x^j)), \\ f_1(x^j) + (x^k - x^j,\ \hat{\nabla} f_1(x^j)) + (h,\ \hat{\nabla} f_1(x^j))\}, \\ \qquad\qquad \delta_1 \geqslant f_1(x^j) \geqslant -\delta_2, \\ f_1(x^j) + (x^k - x^j,\ \hat{\nabla} f_1(x^j)) + (h,\ \hat{\nabla} f_1(x^j)), \quad f_1(x^j) > \delta_1. \end{cases}$$

X. Вычислить следующее приближение

$$x^{k+1} = x^k + h^k.$$

XI. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 2. *Пусть выполняются предположения 2 и шаговые множители* ρ_k, $k = 0, 1, \ldots$, *удовлетворяют условиям*

$$\rho_k \to +0 \quad \textit{при} \quad k \to \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

Тогда бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 2, такова, что*

$$\min_{x \in X^*} \|x^k - x\| \to 0 \quad \textit{при} \quad k \to \infty;$$

$$f_0(x^k) \to \min_{x \in X} f_0(x) \quad \textit{при} \quad k \to \infty.$$

где $X^* = \{x \mid f_0(x) = \min_{y \in X} f_0(y),\ x \in X\}$.

3. Методы проектирования обобщенного градиента для решения предельных экстремальных задач

Задача 3. Найти $\arg\min_{x \in X} f_0(x)$, где $f_0(x) \triangleq \lim_{k \to \infty} \varphi_k(x)$; $\varphi_k : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $k = 0, 1, \ldots$ — заданные функции; $X \subset \boldsymbol{R}^n$.

Предположения 3. (*i*) — функции $\varphi_k(x)$, $k = 0, 1, \ldots$ — выпуклые; (*ii*) — X — выпуклый компакт.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить вектор $\hat{\nabla}\varphi_k(x^k)$ — обобщенный градиент функции $\varphi_k(x)$ в точке x^k.

IV. Найти шаговый множитель ρ_k, удовлетворяющий условиям теоремы 3.

V. Вычислить следующее приближение

$$x^{k+1} = \pi_X(x^k - \rho_k \hat{\nabla}\varphi_k(x^k)).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 3. *Пусть выполняются предположения 3 и* (*iii*) — *последовательность* $\{\varphi_k(x)\}_{k=0}^{\infty}$ *сходится на* X *равномерно к функции* $f_0(x)$; (*iv*) —

$$\rho_k > 0, \quad k = 0, 1, \ldots; \quad \rho_k \to 0 \quad \text{при} \quad k \to \infty; \quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

Тогда для любой сходящейся подпоследовательности $\{x^{k_i}\}_{i=1}^{\infty}$ *последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 3, справедливо предельное соотношение*

$$\lim_{i \to \infty} x^{k_i} = x^* \in X^*,$$

где X^* — *множество решений задачи 3.*

Ниже приводится стохастический аналог алгоритма 3.

Алгоритм 3'

Начало. I. Выбрать произвольное начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализацию ξ^k случайного вектора $\bar{\xi}^k$, условное математическое ожидание которого удовлетворяет соотношению

$$E(\bar{\xi}^k / x^0, x^1, \ldots, x^k) = \hat{\nabla}\varphi_k(x^k).$$

IV. Найти шаговый множитель ρ_k, удовлетворяющий условиям теоремы 3'.

V. Вычислить следующее приближение

$$x^{k+1} = \pi_X(x^k - \rho_k \xi^k).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 3'. *Пусть выполняются все предположения теоремы 3 и условие*

$$\sum_{k=0}^{\infty} (\rho_k)^2 < \infty.$$

Тогда с вероятностью 1 предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, принадлежат множеству решений X^* задачи 3.

Библиографические указания. Пункт 1 написан на основании работ [148, 153, 288], пункт 2 — на основании [126, 129], пункт 3 основан на результатах работы [159].

5.19. Методы условного градиента

1. Реализуемый метод условного градиента

Задача 1. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного выпуклого компактного множества X.

Предположение 1. Функция f_0 — непрерывно дифференцируема.

Метод условного градиента может применяться для решения задачи минимизации нелинейной функции в области, в которой задача минимизации линейной функции решается без особого труда. В данном методе на k-й итерации за направление движения к следующему приближению x^{k+1} выбирается вектор $h^k = z^k - x^k$, где $z^k \in X$ удовлетворяет условию

$$(\nabla f_0(x^k),\ z^k) = \min_{z \in X} (\nabla f_0(x^k),\ z).$$

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Выбрать константу $\beta \in (0,\ 1)$ (рекомендуется $\beta = 1/2$).

III. Положить $k = 0$.

Основной цикл. IV. Вычислить $\nabla f_0(x^k)$.

V. Найти точку $z^k \in X$, удовлетворяющую условию

$$(\nabla f_0(x^k),\ z^k) = \min_{z \in X} (\nabla f_0(x^k),\ z).$$

VI. Вычислить вектор $h^k = z^k - x^k$.

VII. Вычислить $\delta_k = (\nabla f_0(x^k),\ h^k)$.

VIII. Если $\delta_k = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IX.

IX. Положить $j = 0$.

X. Вычислить $\rho_k = \beta^j$.

XI. Если выполняется неравенство

$$f_0(x^k + \rho_k h^k) \leqslant f_0(x^k) + \rho_k (1 - \beta)\, \delta_k,$$

то перейти к шагу XII; иначе положить $j = j + 1$ и перейти к шагу X.

XII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

XIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Если выполнено предположение 1 и* (i) — *множество* X *выпукло и компактно;* (ii) — *градиент функции* f_0 *в области* X *удовлетворяет условию Липшица*

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma \|x - y\|, \quad \gamma < \infty, \quad \forall x, y \in X,$$

то любая предельная точка x^* *бесконечной последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, удовлетворяет необходимому условию минимума функции* f_0 *на множестве* X

$$(\nabla f_0(x^*), x^*) \leqslant (\nabla f_0(x^*), x), \quad \forall x \in X.$$

Условие $\delta_k = 0$ также является необходимым условием минимума функции f_0 на множестве X.

Теорема 1'. *Если выполнены условия теоремы 1 и функция* f_0 *выпукла, то бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, такова, что*

$$\lim_{k \to \infty} f_0(x^k) = \min_{x \in X} f_0(x),$$

причем справедлива следующая оценка скорости сходимости

$$f_0(x^k) - \min_{x \in X} f_0(x) \leqslant \nu_1/k,$$

где $\nu_1 > 0$ — *некоторая константа.*

Теорема 1''. *Если выполнены условия теоремы 1' и* (i) — *область* X *сильно выпукла, т. е. существует число* $\beta_0 > 0$ *такое, что для всех* $x, y \in X$ *точки* $z = (x + y)/2 + w$ *принадлежат* X *для всех* w, *удовлетворяющих условию* $\|w\| \leqslant \beta_0 \|x - y\|^2$; ($ii$) — *существует константа* $\varepsilon_0 > 0$ *такая, что* $\|\nabla f_0(x)\| \geqslant \varepsilon_0$ *для всех* $x \in X$, *то бесконечная последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 1, сходится к точке минимума* x^* *функции* f_0 *на множестве* X *со скоростью геометрической прогрессии*

$$\|x^k - x^*\| \leqslant \nu_2 q_0^k, \quad q_0 < 1,$$

где

$$\nu_2 = ((f_0(x^0) - f_0(x^*))/2\beta_0\varepsilon_0)^{1/2}, \quad q_0 = (1 - \beta_0\varepsilon_0/4\gamma)^{1/2}.$$

2. Алгоритм Франка—Вулфа

Предположения 2. (i) — функция $f_0(x)$ — выпукла и непрерывно дифференцируема; (ii) — X — выпуклое множество.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Положить $k = 0$.

Основной цикл. III. Найти точку $y^k \in X$ такую, что для всех $x \in X$ выполняется неравенство

$$(\nabla f_0(x^k), y^k) \leqslant (\nabla f_0(x^k), x).$$

IV. Найти число $\rho_k \in [0, 1]$ такое, что

$$f_0((1-\rho_k)x^k + \rho_k y^k) \leqslant f_0((1-\rho)x^k + \rho y^k) \text{ для всех } \rho \in [0, 1].$$

V. Положить $x^{k+1} = (1-\rho_k)x^k + \rho_k y^k$.

VI. Если $f_0(x^{k+1}) < f_0(x^k)$, то положить $k = k+1$ и перейти к шагу III; если $f_0(x^{k+1}) = f_0(x^k)$, то положить $x^* = x^{k+1}$ и прекратить вычисления (в этом случае находят решение x^* задачи 1).

Теорема 2. *Пусть выполняются предположения 2. Тогда предельные точки бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порождённой алгоритмом 2, являются решением задачи 1.*

3. Ускоренный алгоритм Франка—Вулфа

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $x^1 \in X$ и целое число $\tau \geqslant 0$.

II. Положить $y^0 = x^1$, $\mu_1^0 = 1$, $k = 1$.

Основной цикл. III. Найти точку $y^k \in X$ такую, что

$$(\nabla f_0(x^k), y^k) \leqslant (\nabla f_0(x^k), x) \text{ для всех } x \in X.$$

IV. Найти число $\rho_k \in [0, 1]$ такое, что

$$f_0((1-\rho_k)x^k + \rho_k y^k) \leqslant f_0((1-\rho)x^k + \rho y^k) \text{ для всех } \rho \in [0, 1].$$

V. Положить $j = 0$.

VI. Вычислить числа

$$\lambda_{k,j}^i = (1-\rho_k)\mu_k^i \text{ для } i = 0, 1, \ldots, k-1;$$
$$\lambda_{k,j}^k = \rho_k.$$

VII. Положить

$$z^{k,j} = (1-\rho_k)x^k + \rho_k y^k.$$

VIII. Если $j \geqslant \tau$, то перейти к шагу XIV; если $j < \tau$, то перейти к шагу IX.

IX. Вычислить множество индексов

$$\mathcal{I}_{k,j} \triangleq \{i \mid (\nabla f_0(z^{k,j}), y^i - z^{k,j}) \leqslant 0, \quad i \in \{0, 1, \ldots, k\}\}.$$

X. Вычислить число $\psi_{k,j}$ и вектор $v^{k,j}$ по формулам

$$\psi_{k,j} = 1/\sum_{i \in \mathcal{I}_{k,j}} \lambda_{k,j}^i; \quad v^{k,j} = \psi_{k,j} \sum_{i \in \mathcal{I}_{k,j}} \lambda_{k,j}^i y^i.$$

XI. Вычислить число $\chi_{k,j} \in [0, 1]$ такое, что

$$f_0((1-\chi_{k,j})z^{k,j} + \chi_{k,j}v^{k,j}) \leqslant f_0((1-\chi)z^{k,j} + \chi v^{k,j})$$
$$\text{для всех } \chi \in [0, 1].$$

XII. Вычислить числа $\lambda_{k\,j+1}^i$, $i = 0, 1, \ldots, k$, по формулам

$$\lambda_{k,j+1}^i = (1 - \chi_{k,j} + \psi_{k,j}\chi_{k,j})\lambda_{k,j}^i \text{ для } i \in \mathcal{I}_{k,j};$$
$$\lambda_{k,j+1}^i = (1-\chi_{k,j})\lambda_{k,j}^i \text{ для } i \notin \mathcal{I}_{k,j}.$$

XIII. Положить $z^{k,j+1} = (1 - \chi_{k,j}) z^{k,j} + \chi_{k,j} v^{k,j}$, $j = j + 1$ и перейти к шагу VIII.

XIV. Положить $x^{k+1} = z^{k,\tau}$ и

$$\mu^{i}_{k+1} = \lambda^{i}_{k,\tau}, \quad i = 0, 1, \ldots, k.$$

XV. Если $f_0(x^{k+1}) < f_0(x^k)$, то положить $k = k + 1$ и перейти к шагу III; иначе положить $x^* = x^{k+1}$ и прекратить вычисления (т. е. в этом случае находят решение x^* задачи 1).

Для алгоритма 3 имеет место теорема, аналогичная теореме 2.

Библиографические указания. При написании пункта 1 использовались работы [320, 130, 222, 444], пункты 2 и 3 написаны на основании работ [521, 420].

В работе [424] можно найти дополнительные сведения об алгоритмах Франка — Вулфа.

5.20. Методы сопряженных градиентов

1. Метод сопряженных градиентов

Задача 1. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X \triangleq \{x \mid f_j(x) = 0, \quad j = 1, \ldots, m, \quad x \in R^n\}.$$

Предположения 1. (*i*) — *функции* $f_i(x)$, $i = 0, 1, \ldots, m$ — дважды непрерывно дифференцируемы в окрестности некоторой точки x^*, в которой выполняются достаточные условия локального минимума

$$f_j(x^*) = 0, \quad j = 1, \ldots, m; \quad \nabla f_0(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla f_j(x^*) = 0;$$

$$\left((\nabla^2_{xx} f_0(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla^2_{xx} f_j(x^*))\, h, h\right) \geqslant \alpha \|h\|^2, \quad \alpha > 0;$$

$$(\nabla f_j(x^*), h) = 0.$$

Для применения метода сопряженных градиентов к задачам условной минимизации необходимо иметь хорошее начальное приближение точки локального минимума x^*. По сути, приводимый здесь метод является методом сопряженных градиентов, примененным для безусловной минимизации некоторой вспомогательной функции. На каждой итерации алгоритма требуется лишь вычислять градиенты функций $f_j(x)$, $j = 0, 1, \ldots$, и решать задачу минимизации одномерной функции. Алгоритм 1 строит последовательность $\{x^k\}_{k=0}^{\infty}$, локально сходящуюся к x^* с квадратичной скоростью.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение x^0 из окрестности точки локального минимума x^*.

II. Положить $k = 0$.

Основной цикл. III. Вычислить векторы $\nabla f_0(x^k)$, $\nabla f_1(x^k)$, ..., $\nabla f_m(x^k)$.

IV. Положить $i = 1$.

V. Положить $\overset{0}{e}_{(j)} = \nabla f_j(x^k)$, $j = 1, \dots, m$.

VI. Вычислить векторы $\overset{i}{e}_{(j)}$, $j = 1, \dots, m$ по правилам

$$\overset{i}{e}_{(i)} = (\overset{i-1}{e}_{(i)}, \overset{0}{e}_{(i)})^{-1} \overset{i-1}{e}_{(i)};$$

$$\overset{i}{e}_{(j)} = \overset{i-1}{e}_{(j)} - (\overset{i-1}{e}_{(j)}, \overset{0}{e}_{(i)}) \overset{i}{e}_{(i)}, \quad j = 1, \dots, m; \quad j \neq i.$$

VII. Если $i = m$, то положить $e^j = \overset{m}{e}_{(j)}$, $j = 1, \dots, m$, и перейти к шагу VIII; иначе (если $i < m$) положить $i = i + 1$ и перейти к шагу VI.

VIII. Вычислить числа

$$\bar{\lambda}_j = -(\nabla f_0(x^k), e^j), \quad j = 1, \dots, m.$$

IX. Определить функцию

$$\varphi(x) = f_0(x) + \sum_{j=1}^{m} \bar{\lambda}_j f_j(x).$$

X. Положить $s = 0$.

XI. Вычислить n-мерные векторы

$$u^0 = x^k - \sum_{j=1}^{m} f_j(x^k) e^j;$$

$$g^0 = \nabla\varphi(u^0) - \sum_{j=1}^{m} (\nabla\varphi(u^0), e^j) \nabla f_j(x^k);$$

$$h^0 = -g^0.$$

XII. Если $s = n - m$, то перейти к шагу XVI; иначе (если $s < n - m$) перейти к шагу XIII.

XIII. Найти число $\rho_s \geqslant 0$ из условия

$$\varphi(u^s + \rho_s h^s) = \min_{\rho \geqslant 0} \varphi(u^s + \rho h^s).$$

XIV. Вычислить векторы u^{s+1}, g^{s+1}, h^{s+1}:

$$u^{s+1} = u^s + \rho_s u^s;$$

$$g^{s+1} = \nabla\varphi(u^{s+1}) - \sum_{j=1}^{m} (\nabla\varphi(u^{s+1}), e^j) \nabla f_j(x^k);$$

$$h^{s+1} = -g^{s+1} + (\|g^{s+1}\| / \|g^s\|)^2 h^s.$$

XV. Положить $s = s + 1$ и перейти к шагу XII.

XVI. Положить $x^{k+1} = u^{n-m}$, $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются предположения 1 и (ii) — вторые производные функций $f_0(x)$, $f_1(x)$, ..., $f_m(x)$ удовлетворяют в*

окрестности точки x^* условию Липшица; (iii) — градиенты $\nabla f_1(x^*)$, ..., $\nabla f_m(x^*)$ — линейно-независимы.

Тогда последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, локально сходится к точке x^* с квадратичной скоростью, т. е.

$$\|x^k - x^*\| \leqslant \beta^{-1} (\beta \|x^0 - x^*\|)^{2^k}, \quad k = 0, 1, \ldots,$$

где $\beta > 0$ — некоторая константа.

2. Стохастический аналог метода сопряженных градиентов

Задача 2. Найти $\arg\min_{x \in X} f_0(x)$.

Предположения 2. (*i*) — функция $f_0: R^n \to R^1$ — непрерывно дифференцируема и выпукла вниз; (*ii*) — X — выпуклое, ограниченное и замкнутое множество.

Стохастический аналог метода сопряженных градиентов может быть применен и в том случае, когда функция (или ее градиент) точно не вычисляются. На каждой итерации строится вектор стохастического квазиградиента ξ^k и направление z^{k+1} движения к следующему приближению x^{k+1} выбирается с помощью определенного осреднения векторов ξ^k, ξ^{k-1}, ξ^{k-2} .... Осреднение реализуется таким образом, чтобы обеспечить (с ростом k) приближение значения z^k к градиенту минимизируемой функции в точке x^k.

Алгоритм 2

Начало. I. Выбрать: произвольное начальное приближение $x^0 \in R^n$; шаговые множители $\rho_1(0)$ и $\rho_2(0)$, удовлетворяющие условиям теоремы 2; произвольный вектор $z^0 \in R^n$ и произвольное число $\delta \geqslant \|z^0\|$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить реализацию ξ^k случайного вектора $\bar{\xi}^k$, условное математическое ожидание которого

$$E(\bar{\xi}^k/(x^0, z^0), \ldots, (x^k, z^k)) = \alpha \nabla f_0(x^k) + b^k, \tag{5.80}$$

где α — некоторое неотрицательное число; b^k — вектор «ошибки», измеримый относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами (x^0, z^0), ..., (x^k, z^k).

IV. Вычислить вектор

$$\bar{z}^{k+1} = z^k + \rho_2(k)(\xi^k - z^k).$$

V. Вычислить вектор

$$z^{k+1} = \begin{cases} \bar{z}^{k+1}, & \text{если } \|\bar{z}^{k+1}\| \leqslant \delta; \\ \delta \bar{z}^{k+1}/\|\bar{z}^{k+1}\|, & \text{если } \|\bar{z}^{k+1}\| > \delta. \end{cases}$$

VI. Вычислить следующее приближение

$$x^{k+1} = \pi_X(x^k - \rho_1(k) z^{k+1}).$$

VII. Вычислить значения шаговых множителей $\rho_1(k+1)$ и $\rho_2(k+1)$, удовлетворяющие условиям теоремы 2.

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполнены предположения 2 и условия:* (*i*) — *градиенты функции* f_0 *удовлетворяют условию Липшица*

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \bar{\beta}_1 \|x - y\|, \quad \bar{\beta}_1 < \infty;$$

(*ii*) —

$$E(\|\bar{\xi}^k\|^2/(x^0, z^0), \ldots, (x^k, z^k)) \leqslant \beta_2 < \infty, \quad k = 0, 1, \ldots;$$

(*iii*) — $\rho_1(k)$, $\rho_2(k)$, b^k — *измеримы относительно σ-подалгебры* $\mathcal{B}_k$ *и такие, что*

$$\sum_{k=0}^{\infty} E\rho_1^2(k) < \infty, \quad \sum_{k=0}^{\infty} E\rho_1(k)\|b^k\| < \infty;$$

$$\sum_{k=0}^{\infty} E\rho_2^2(k) < \infty, \quad \sum_{k=0}^{\infty} \rho_1(k) = \infty \text{ п. н};$$

(*iv*) — *для некоторых чисел* λ_k *выполняются условия*

$$\rho_1(k) + \rho_2(k)\|b^k\| - \lambda_k\rho_2(k) \leqslant 0 \text{ п. н};$$

$$\sum_{k=0}^{\infty} E\lambda_k(\rho_1(k) + \rho_2(k)\|b^k\|) < \infty.$$

Тогда последовательность $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 2, почти наверное сходится к решению задачи 2 (а если решение не единственно, то* $\{x^k\}_{k=0}^{\infty}$ *сходится к одному из решений).*

Замечание 2. В случае $\|b^k\| \equiv 0$ условия теоремы 2 (*iii*) и (*iv*) выполняются, если только выбрать

$$\rho_1(k) = k^{\tau+}, \quad \rho_2(k) = k^{\varepsilon}; \quad \lambda_k = k^{\tau},$$

где числа τ и ε определяются неравенствами

$$-1 < \varepsilon < -1/2;$$

$$-(1+\varepsilon) \leqslant \tau < -(1+\varepsilon)/2.$$

Библиографические указания. Пункт 1 написан на основании работы [232], пункт 2 основан на результатах работы [98].

5.21. Методы управляющих последовательностей

Задача 0. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j \in \mathcal{J}, \quad x \in R^n\}, \quad \mathcal{J} = \{1, \ldots, m\}.$$

Предположения 0. (*i*) — функции f_0 и f_j, $j \in \mathcal{J}$ — выпуклые; (*ii*) — существует точка $x^* = \arg\min_{x \in X} f_0(x)$ и градиенты $\nabla f_j(x^*)$,

$j \in \mathcal{J}(x^*)$ — линейно независимы, где $\mathcal{J}(x^*) = \{j \mid f_j(x^*) = 0,\ j \in \in \mathcal{J}\}$ (в дальнейшем предполагается, что $\mathcal{J}(x^*) = \{1, \ldots, r\}$); (*iii*) — дифференциальное преобразование $\left(\frac{\partial}{\partial x} f_j(x)\right)$, $j = 1, \ldots, r$, непрерывно в области $W(x^*, \delta) = \{x \mid \| x - x^* \| < \delta\}$, где $\delta > 0$ — достаточно малое число.

В методе управляющих последовательностей на основе методов безусловной оптимизации строят операторы релаксации на нелинейном многообразии в окрестности решения x^*, которые порождают последовательности приближений $\{x^k\}_{k=0}^{\infty}$, локально сходящиеся к решению x^* с быстротой соответствующих процессов безусловной оптимизации. В начальной фазе различные конкретизации общего метода управляющих последовательностей совпадают с процессами оптимизации при наличии ограничений, а в заключительной фазе — с процессами безусловной оптимизации.

1. Общий метод минимизации при наличии ограничений

Различные методы оптимизации при наличии ограничений являются конкретизациями общего алгоритма.

Алгоритм 1

Начало. I. Определить на $X \times R_+^1$ неотрицательную полунепрерывную снизу в точках $(x, 0)$, $x \in X$, функцию $\mu(x, \delta)$ такую, что из $\mu(x, 0) = 0$ следует $x = x^*$ (здесь δ — достаточно малое число).

II. Определить оператор $C : X \to X$, удовлетворяющий условию

$$f_0(x) - f_0(Cx) \geqslant \sigma(\delta) > 0, \quad \forall x \in \{x \mid \mu(x, \varepsilon) \geqslant \delta,\ 0 \leqslant \varepsilon \leqslant \delta\}. \quad (5.81)$$

III. Найти начальное приближение $x^0 \in X$.

IV. Выбрать управляющую последовательность $\{\delta_i\}_{i=0}^{\infty}$, т. е. монотонно убывающую последовательность неотрицательных чисел, имеющую своим пределом нуль.

V. Положить $k = 0$, $i(0) = 0$.

Основной цикл. VI. Если выполняется неравенство $\mu(x^k, \delta_{i(k)}) > \delta_{i(k)}$, то положить $x^{k+1} = Cx^k$, $i(k+1) = i(k)$ и перейти к шагу VIII; если $0 < \mu(x^k, \delta_{i(k)}) \leqslant \delta_{i(k)}$, то положить $x^{k+1} = Cx^k$, $i(k+1) = i(k) + 1$ и перейти к шагу VIII, а если $\mu(x^k, \delta_{i(k)}) = 0$, то вычислить $\mu(x^k, 0)$ и перейти к шагу VII.

VII. Если $\mu(x^k, 0) = 0$, то положить $x^* = x^k$ и прекратить вычисления (в этом случае решение задачи 0 находят за конечное число итераций); иначе положить $x^{k+1} = Cx^k$, $i(k+1) = i(k) + 1$ и перейти к шагу VIII.

VIII. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 1. *Если выполняются предположения 0, а последовательность $\{x_k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, такова, что последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ ограничена снизу, то любая сходящаяся подпоследовательность $\{x^{k_l}\}_{l=0}^{\infty}$ имеет пределом x^*; если допол-*

нительно выполняется условие

$$f_0(Cx) \leqslant f_0(x), \quad \text{\textit{то}} \quad \lim_{k\to\infty} f_0(x^k) = f_0(x^*),$$

и если, кроме того, точка x^* *единственная, то*

$$\lim_{k\to\infty} x^k = x^*.$$

Ниже приводятся примеры определения операторов C и функций $\mu(x, \delta)$.

Пример 1

$$\mu(x, \delta) = -\min\{(\nabla f_0(x), \xi) \mid (\nabla f_j(x), \xi) \leqslant -\delta,$$
$$j \in \mathcal{J}(x, \delta); \quad |\xi_j| \leqslant 1, \quad j = 1, \ldots, n\} = -(\nabla f_0(x), \xi(x)),$$

где

$$\mathcal{J}(x, \delta) = \{j \mid -\delta < f_j(x) \leqslant 0, \quad j \in \mathcal{J}\};$$
$$Cx = x + t\xi(x) \quad (t > 0).$$

Пример 2

$$\mu(x, \delta) = \Big\| \sum_{j \in \mathcal{J}'(x,\delta)} v_j(x) \nabla f_j(x) \Big\| =$$
$$= \min\Big\{\Big\| \sum_{j \in \mathcal{J}'(x,\delta)} v_j \nabla f_j(x) \Big\| \,\Big|\, \sum_{j \in \mathcal{J}'(x,\delta)} v_j = 1, \ v_j \geqslant 0, \ j \in \mathcal{J}'(x, \delta)\Big\},$$

где

$$\mathcal{J}'(x, \delta) = \mathcal{J}(x, \delta) \cup \{0\};$$
$$Cx = x + t\xi(x), \quad \xi(x) = \sum_{j \in \mathcal{J}'(x,\delta)} v_j(x) \nabla f_j(x).$$

Если в алгоритме 1 функцию μ и оператор C определять, как в примере 1, то алгоритм 1 превращается в одну из версий метода возможных направлений; если μ и C определять, как в примере 2, то придем к варианту метода одновременного решения пары двойственных задач выпуклого программирования.

2. Метод управляющих последовательностей

Ниже приводится алгоритм, синтезирующий методы условной и безусловной оптимизации: в начальной стадии алгоритма используется оператор C, а начиная с некоторого момента вместо оператора C используется оператор релаксации $\tilde{T}$.

Определение 2. Оператором релаксации $\tilde{T}$: $W(x^*, \delta) \to W(x^*, \delta)$ называется такой оператор, для которого выполняется

$$\|\tilde{T}x - x^*\| \leqslant \|x - x^*\|.$$

Методы безусловной оптимизации порождают операторы релаксации, обладающие следующими свойствами:

$$\|\tilde{T}^k x - x^*\| < \alpha_1 q^k, \quad q < 1; \tag{5.82}$$

$$\|\tilde{T}^k x - x^*\| \leqslant \prod_{l=1}^{k} q_l, \quad q_l \to 0; \tag{5.83}$$

$$\|\tilde{T}^k x - x^*\| \leqslant \alpha_2 q^{2^k}, \quad q < 1. \tag{5.84}$$

Приведем примеры модифицированных операторов релаксации на многообразие активных ограничений, сохраняющих свойства (5.82) и (5.84).

Пример 3

Модифицированный оператор градиентной релаксации

$$\tilde{T}x = (z - tG(z, P_f\, y),\ P_f y - tY(z, P_f y)\, G(z, P_f y)),$$

где

$$x = (z, y) \in \boldsymbol{R}^n, \quad z \in \boldsymbol{R}^{n_1}, \quad y \in \boldsymbol{R}^r, \quad n_1 + r = n;$$

$$f(x) = f(z, y) = (f_1(x), \ldots, f_r(x));$$

$$\frac{\partial f(x)}{\partial x} = \left(\frac{\partial}{\partial x} f_j(x)\right)_{j=1,\ldots,r} = \left(\frac{\partial}{\partial z} f(x), \frac{\partial}{\partial y} f(x)\right);$$

$$P_f\, y = y - \left[\frac{\partial}{\partial y} f(z, y)\right]^{-1} f(z, y);$$

$$Y(x) = Y(z, y) = -\left[\frac{\partial}{\partial y} f(x, y)\right]^{-1} \frac{\partial}{\partial z} f(z, y);$$

$$G(x) = \frac{\partial}{\partial z} f_0(x) + \frac{\partial}{\partial y} f_0(x)^T Y(x), \quad t > 0.$$

Модифицированный оператор градиентной релаксации сохраняет свойство (5.82).

Пример 4

Модифицированный оператор Ньютона

$$\tilde{T}x = (z - [H(z, P_f y)]^{-1} G(z, P_f y),\ P_f y - Y(z, P_f y) \times$$
$$\times [H(z, P_f y)]^{-1} G(z, P_f y)),$$

где G, P_f, Y определены, как в примере 3, и матрица $H(x)$ вычисляется по формуле

$$H(x) = \frac{\partial^2}{\partial z^2} f_0(x) + \frac{\partial^2}{\partial z \partial y} f_0(x) \frac{\partial y(z)}{\partial z} + \left(\frac{\partial y(z)}{\partial z}\right)^T \times$$
$$\times \left(\frac{\partial^2}{\partial y \partial z} f_0(x) + \frac{\partial^2}{\partial y^2} f_0(x) \frac{\partial y(z)}{\partial z}\right) + \sum_{l=1}^{r} \left(\frac{\partial}{\partial y_l} f_0(x) \frac{\partial^2 y_l(z)}{\partial z^2}\right),$$

где $y(z) = (y_1(z), \ldots, y_r(z))$ — вектор-функция, определяемая неявной системой

$$f(x) = f(z, y) = 0,$$

а матрица $\partial y(z)/\partial z$ вычисляется по формуле

$$\frac{\partial y(z)}{\partial z} = -\left(\frac{\partial}{\partial y} f(x)\right)^{-1} \frac{\partial}{\partial z} f(x).$$

Модифицированный оператор Ньютона сохраняет свойство (5.84).

Алгоритм 2

Начало. I. Выбрать оператор C, удовлетворяющий условию (5.82); оператор релаксации $\tilde{T}$; функцию $\mu(x, \delta)$, определенную на шаге I алгоритма 1.

II. Выбрать произвольную управляющую последовательность $\{\delta_i\}_{i=0}^{\infty}$ и согласованную с оператором $\tilde{T}$ управляющую последовательность $\{\lambda_k\}_{k=0}^{\infty}$.

(Управляющая последовательность $\{\lambda_k\}_{k=0}^{\infty}$ согласована с оператором $\tilde{T}$, если

$$\lim_{k\to\infty} \nu(\tilde{T}^k x)\lambda_k^{-1} = 0 \quad \forall x \in W(x^*, \delta),$$

где функция $\nu(x)$, характеризующая меру неоптимальности вектора x, принадлежащего многообразию активных ограничений, определяется по правилам

$$\nu(x) = \max\{\|G(x)\|, v(x), \psi(x)\},$$

здесь $G(x)$ определяется в примере 3;

$$\psi(x) = \max\{f_j(x) \mid j \in \mathcal{J}\};$$

$$v(x) = \max\{-u_j(x) \mid j \in \mathcal{J}(x^*)\}, \ (u_j(x), \ j \in \mathcal{J}(x^*)) = -\frac{\partial}{\partial y} f_0(x) \times \\ \times \left(\frac{\partial}{\partial y} f(x)\right)^{-1}.$$

Примером последовательностей, согласованных с оператором $\tilde{T}$, обладающего свойствами (5.82) — (5.84), являются следующие:

$$\{(1+k)^{-1}\}_{k=0}^{\infty}; \quad \{q^k\}_{k=0}^{\infty}, \quad q<1; \quad \{(k+1)^{-k}\}_{k=0}^{\infty}).$$

III. Выбрать $\sigma \in (0, 1)$.

IV. Выбрать произвольное начальное приближение $x^0 \in X$.

V. Положить $k = 0$, $i(0) = 0$.

Основной цикл. VI. Если выполняется неравенство $\mu(x^k, \delta_{i(k)}) > \delta_{i(k)}$, то перейти к шагу VII, если $0 < \mu(x^k, \delta_{i(k)}) \leqslant \delta_{i(k)}$, то перейти к шагу VIII, если $\mu(x^k, \delta_{i(k)}) = 0$, то перейти к шагу XVIII.

VII. Положить $x^{k+1} = Cx^k$, $i(k+1) = i(k)$ и перейти к шагу XIX.

VIII. Вычислить значение $\tilde{\mu}(x^k, \delta_{i(k)})$, где функция $\tilde{\mu}(x, \delta)$ определяется по формуле

$$\tilde{\mu}(x, \delta) = -\min\{(\nabla f_0(x), \xi) \mid (\nabla f_j(x), \xi) \leqslant \delta, \ j \in \mathcal{J}(x, \delta); \\ |\xi_i| \leqslant 1, \quad i = 1, \ldots, n\},$$

где множество

$$\mathcal{J}(x,\ \delta)=\{j\,|-\delta\leqslant f_j(x)\leqslant 0,\quad j\in\mathcal{J}\}.$$

IX. Вычислить

$$\varepsilon_k=[\tilde{\mu}(x^k,\quad \delta_{i(k)})]^{(1-\sigma)/2}.$$

X. Найти множество индексов

$$\mathcal{J}(x^k,\ \varepsilon_k)=\{j\,|-\varepsilon_k\leqslant f_j(x^k)\leqslant 0,\quad j\in\mathcal{J}\}.$$

XI. Положить

$$f(x)=(f_j(x),\quad j\in\mathcal{J}(x^k,\ \varepsilon_k)).$$

XII. Вычислить величину $\left|\det\frac{\partial f(x^k)}{\partial y}\right|$ и положить

$$\Delta(x^k)=\left|\det\frac{\partial f(x^k)}{\partial y}\right|;$$

если $\Delta(x^k)\leqslant\delta_{i(k)}$, то перейти к шагу XIII; если $\Delta(x^k)>\delta_{i(k)}$, то перейти к шагу XIV.

XIII. Положить

$$x^{k+1}=Cx^k,\quad i(k+1)=i(k)+1$$

и перейти к шагу XIX.

XIV. Положить $\bar{x}^0=x^k$.

XV. Положить $l=0$.

XVI. Если $\nu(\bar{x}^l)\leqslant\lambda_l$, то положить $\bar{x}^{l+1}=\tilde{T}\bar{x}^l$ и перейти к шагу XVII, если $\nu(\bar{x}^l)>\lambda_l$, то положить

$$x^{k+1}=\arg\min\{f_0(x)\,|\,x\in\{\bar{x}^0,\ \ldots,\ \bar{x}^l\}\};$$
$$i(k+1)=i(k)+l+1$$

и перейти к шагу XIX.

XVII. Положить $l=l+1$ и перейти к шагу XVI.

XVIII. Если $\mu(x^k,\ 0)=0$, то положить $x^*=x^k$ и прекратить вычисления; иначе перейти к шагу VIII.

XIX. Положить $k=k+1$ и перейти к шагу VI.

Теорема 2. *Если выполнены предположения 0 и (iv) — одна из функций $f_j(x)$, $j\in\mathcal{J}'_+(x^*)$, сильно выпукла; (v) — функции $f_j(x)$, $j\in\mathcal{J}\cup\{0\}$, достаточно гладки, то в алгоритме 2 последовательность приближений, начиная с некоторого момента, порождается лишь оператором $\tilde{T}$.*

Здесь $\mathcal{J}'_+(x^*)=\{j\,|\,u_j^*>0,\ j\in\mathcal{J}(x^*)\}\cup\{0\}$, где u_j^* — множители в соотношении Куна — Таккера

$$\nabla f_0(x^*)+\sum_{j\in\mathcal{J}(x^*)}u_j^*\nabla f_j(x^*)=0,\quad u_j^*\geqslant 0,\quad j\in\mathcal{J}(x^*).$$

Библиографические указания. При написании параграфа использованы работы [304, 305, 306].

5.22. Методы внутренней аппроксимации

Задача 1. Найти $\arg\min\limits_{x\in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и заданного множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j = 1, \dots, m, \quad x \in R^n\}.$$

Предположения 1. (*i*) — функции f_j, $j = 0, 1, \dots, l$ — дифференцируемые и выпуклые; (*ii*) — функции f_j, $j = l + 1, l + 2, \dots, m$ — дифференцируемые; (*iii*) — множество X компактное.

Метод внутренней аппроксимации сводится к решению последовательности аппроксимирующих задач выпуклого программирования. В k-й аппроксимирующей задаче каждое невыпуклое ограничение $f_j(x) \leqslant 0$, $j = l + 1, \dots, m$, заменяется специально построенным выпуклым ограничением. Алгоритм внутренней аппроксимации прекращает вычисления, когда условия Куна — Таккера, связанные с аппроксимирующей выпуклой задачей, соответствуют условиям Куна — Таккера задачи 1.

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Положить $\sigma_0 = f_0(x^0)$.

III. Найти множество

$$Z_0 = \{x \mid \sigma_0 = f_0(x), \quad x \in X\}.$$

IV. Положить $k = 1$

Основной цикл. V. Выбрать точку $x^k \in Z_{k-1}$ и построить функции $\bar{f}_j(x, x^k)$, $j = l + 1, \dots, m$, обладающие следующими свойствами:

а) $\bar{f}_j(x, x^k)$, $j = l + 1, \dots, m$ — дифференцируемые выпуклые функции;

б) $f_j(x) \leqslant \bar{f}_j(x, x^k)$ для всех точек $x \in X_k$, где

$$X_k = \{x \mid f_j(x) \leqslant 0, \ j = 1, \dots, l;$$
$$\bar{f}_j(x, x^k) \leqslant 0, \quad j = l + 1, \dots, m; \quad x \in R^n\};$$

в) $f_j(x^k) = \bar{f}_j(x^k, x^k), \quad j = l + 1, \dots, m;$

г) $\partial f_j(x^k)/\partial x_i = \partial \bar{f}_j(x^k, x^k)/\partial x_i$, $i = 1, \dots, n$, $j = 1 + l, \dots, m$;

д) множество X_k удовлетворяет условию регулярности Слейтера для задач выпуклого программирования.

VI. Решить аппроксимирующую задачу выпуклого программирования:

найти $\arg\min\limits_{x\in X_k} f_0(x)$ для заданной выпуклой функции $f_0 : R^n \to R^1$ и заданного выпуклого множества

$$X_k = \{x \mid f_j(x) \leqslant 0, \quad j = 1, \dots, l;$$
$$\bar{f}_j(x, x^k) \leqslant 0, \quad j = l + 1, \dots, m, \quad x \in R^n\}$$

и положить $\sigma_k = \min\limits_{x\in X_k} f_0(x)$.

VII. Если $\sigma_k = \sigma_{k-1}$, то прекратить вычисления (в этом случае x^k — решение для исходной задачи 1 в смысле Куна — Таккера); иначе найти множество

$$Z_k = \{x \mid \sigma_k = f_0(x), \quad x \in X_k\}$$

и перейти к шагу VIII.

VIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. Если выполняются предположения 1, то алгоритм 1 либо останавливается за конечное число итераций в точке Куна — Таккера, либо порождает бесконечную последовательность $\{x^k\}_{k=0}^{\infty}$, *предельные точки которой являются точками Куна — Таккера.*

Следствие 1. Если точка x^k получена как решение алгоритма внутренней аппроксимации за конечное число итераций и x^k — внутренняя точка множества X_k, то x^k является точкой локального минимума в задаче 1.

Замечание 1. Алгоритм внутренней аппроксимации может быть использован для решения задач нелинейного программирования с невыпуклой целевой функцией f_0. Для этого целевую функцию f_0 заменяют новой переменной x_{n+1} (выпуклой функцией), а ограничение $f_0(x) - x_{n+1} \leqslant 0$ добавляют к ограничениям задачи. Для решения новой задачи в пространстве R^{n+1} применяют алгоритм 1.

Библиографические указания. Параграф написан на основании работы [515]. В работе [533] используются аппроксимирующие касательные плоскости к целевой функции для сведения исходной задачи нелинейного программирования к решению последовательности более простых задач оптимизации.

5.23. Методы покоординатного спуска

Задача 0. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и параллелепипеда

$$X = \{x \mid \alpha_i \leqslant x_i \leqslant \beta_i, \quad i = 1, \ldots, n, \quad x \in R^n\},$$

где α_i, β_i, $i = 1, \ldots, n$ — заданные действительные числа.

Предположения 0. (i) — функция $f_0(x)$ — непрерывно дифференцируема и для любых x, $y \in X$ выполняется

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma \|x - y\|, \quad 0 < \gamma < \infty.$$

1. Детерминированный покоординатный спуск

В методе детерминированного покоординатного спуска за направление движения последовательно выбираются орты. Параметры метода ρ и δ вычисляются в процессе работы алгоритма. Алгоритм включает «большие» итерации по индексу k и «малые» — по индексу j, причем при каждом k «малые» итерации заканчиваются при конечном значении индекса j.

Алгоритм 1

Начало. 1. Выбрать произвольное начальное приближение $x^0 \in X$ и вычислить $f_0(x^0)$.

II. Выбрать произвольные константы ρ_0, δ_0, w, удовлетворяющие неравенствам

$$\rho_0 > 0; \quad \delta_0 > 0; \quad w > 1.$$

III. Задать векторы $e^1, \ldots, e^n$ (здесь e^i, $i = 1, \ldots, n$ — вектор, i-я координата которого равна единице, а остальные равны нулю).

IV. Положить $k = 0$.

Основной цикл. V. Положить $j = 0$.

VI. Вычислить индекс i_k ($i_k \in [1 : n]$) по правилу

$$i_k = k - n\,\mathrm{Ent}\,(k/n) + 1,$$

где Ent (t) — целая часть числа t.

VII. Положить $h^k = e^{i_k}$.

VIII. Вычислить векторы

$$x^{k(+)} = x^k + (\rho_k/2^j)\,h^k; \quad x^{k(-)} = x^k - (\rho_k/2^j)\,h^k.$$

IX. Если $x^{k(+)} \in X$ и $x^{k(-)} \in X$, то перейти к шагу X; если $x^{k(+)} \in X$ и $x^{k(-)} \notin X$, то вычислить $f_0(x^{k(+)})$ и перейти к шагу XV; если $x^{k(+)} \notin X$ и $x^{k(-)} \in X$, то вычислить $f_0(x^{k(-)})$ и перейти к шагу XVI; если $x^{k(+)} \notin X$ и $x^{k(-)} \notin X$, то положить $j = j + 1$ и перейти к шагу VIII.

X. Вычислить $f_0(x^{k(+)})$.

XI. Если выполняется неравенство

$$f_0(x^{k(+)}) - f_0(x^k) \leqslant -\rho_k\delta_k/2^j, \tag{5.85}$$

то положить

$$x^{k+1} = x^{k(+)}; \quad \rho_{k+1} = \rho_k; \quad \delta_{k+1} = \delta_k; \quad f_0(x^{k+1}) = f_0(x^{k(+)})$$

и перейти к шагу XVIII; иначе перейти к шагу XII.

XII. Вычислить $f_0(x^{k(-)})$.

XIII. Если выполняется неравенство

$$f_0(x^{k(-)}) - f_0(x^k) \leqslant -\rho_k\delta_k/2^j, \tag{5.86}$$

то положить

$$x^{k+1} = x^{k(-)}; \quad \rho_{k+1} = \rho_k; \quad \delta_{k+1} \leqslant \delta_k; \quad f_0(x^{k+1}) = f_0(x^{k(-)})$$

и перейти к шагу XVIII; иначе перейти к шагу XIV.

XIV. Если выполняется неравенство

$$|f_0(x^{k(+)}) - f_0(x^{k(-)})| \leqslant 2\rho_k w\delta_k/2^j,$$

то перейти к шагу XVII; иначе положить $j = j + 1$ и перейти к шагу VIII.

XV. Если выполняется неравенство (5.85), то положить

$$x^{k+1} = x^{k(+)}; \quad f_0(x^{k+1}) = f_0(x^{k(+)}); \quad \rho_{k+1} = \rho_k; \quad \delta_{k+1} = \delta_k$$

и перейти к шагу XVIII; иначе перейти к шагу XVII.

XVI. Если выполняется неравенство (5.86), то положить

$$x^{k+1} = x^{k(-)}; \quad f_0(x^{k+1}) = f_0(x^{k(-)}); \quad \rho_{k+1} = \rho_k; \quad \delta_{k+1} = \delta_k$$

и перейти к шагу XVIII; иначе перейти к шагу XVII.

XVII. Если выполняются равенства

$$i_k = n; \quad x^k = x^{k-n+1},$$

то положить

$$x^{k+1} = x^k; \quad f_0(x^{k+1}) = f_0(x^k); \quad \rho_{k+1} = \rho_k/2; \quad \delta_{k+1} = \delta_k/2$$

и перейти к шагу XVIII; если $i_k \neq n$ или $x^k \neq x^{k-n+1}$, то положить

$$x^{k+1} = x^k; \quad f_0(x^{k+1}) = f_0(x^k); \quad \rho_{k+1} = \rho_k; \quad \delta_{k+1} = \delta_k$$

и перейти к шагу XVIII.

XVIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если выполнены предположения 0, функция $f_0(x)$ выпукла на X и множество*

$$X_0 = \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in X\}$$

ограничено (здесь x^0 — начальное приближение в алгоритме 1), то алгоритм 1 порождает последовательность $\{x^k\}_{k=0}^{\infty}$, для которой

$$\lim_{k\to\infty} \inf_{x^* \in X^*} \| x^k - x^* \| = 0,$$

где

$$X^* = \{x^* \mid f_0(x^*) = \min_{x \in X} f_0(x), \quad x^* \in X\}.$$

2. Случайный покоординатный спуск

В методе случайного покоординатного спуска за направление движения в k-й итерации случайно выбирается i_k-й орт. Для вычисления шаговых множителей ρ_k приводятся три различные правила, легко реализуемые на практике.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Задать векторы $e^1, \ldots, e^n$ (здесь e^i, $i = 1, \ldots, n$ — i-й координатный вектор).

III. Положить $k = 0$.

Основной цикл. IV. Найти независимую реализацию i_k случайной величины $\tilde{i}$, которая принимает значения из множества $\{1, \ldots, n\}$, соответственно, с вероятностями $p_1 = 1/n, \ldots, p_n = 1/n$.

V. Вычислить частную производную $\frac{\partial}{\partial x_{i_k}} f_0(x^k)$.

VI. Вычислить шаговый множитель

$$\rho_k = \begin{cases} -\min\left\{x_{i_k}^k - \alpha_{i_k}, \ \frac{\partial}{\partial x_{i_k}} f_0(x^k)/\gamma\right\}, & \text{если } \frac{\partial}{\partial x_{i_k}} f_0(x^k) \geqslant 0; \\ -\min\left\{\beta_{i_k} - x_{i_k}^k, \ \left|\frac{\partial}{\partial x_{i_k}} f_0(x^k)\right| / \gamma\right\}, & \text{если } \frac{\partial}{\partial x_{i_k}} f_0(x^k) \leqslant 0. \end{cases}$$

Здесь предполагается, что известна константа Липшица γ в условии (*i*) предположения 0. Если константа γ неизвестна, то ρ_k вычисляется по формулам, приведенным в замечании 2.

VII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k e^{i_k}.$$

VIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Если выполняется предположение 0 и множество*

$$X_0 = \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in X\}$$

ограничено (здесь x^0 — начальное приближение в алгоритме 2), то алгоритм 2 порождает последовательность $\{x^k\}_{k=0}^{\infty}$, для которой с вероятностью 1

$$\lim_{k \to \infty} \inf_{x^* \in X^*} \| x^k - x^* \| = 0,$$

где

$$X^* (\triangleq \{x^* \mid (\nabla f_0(x^*), \ x - x^*) \geqslant 0 \text{ для всех } x \in X \}$$

— множество стационарных точек функции $f_0(x)$. Если, кроме того, функция $f_0(x)$ выпукла на X, то с вероятностью 1

$$\lim_{k \to \infty} f_0(x^k) = \min_{x \in X} f_0(x).$$

Замечание 2. Шаговый множитель ρ_k на шаге VI алгоритма **2** можно выбирать из условия

$$f_0(x^k - \rho_k e^{i_k}) \leqslant \lambda_k f_0(x^k) + (1 - \lambda_k) w_k,$$

где $0 < \lambda \leqslant \lambda_k \leqslant 1; \quad w_k = \inf_{\rho} f_0(x^k - \rho e^{i_k}); \quad x^k - \rho e^{i_k} \in X.$

В качестве ρ_k также может быть выбрано любое число

$$\rho \geqslant \delta \mu_k,$$

где $\delta \in (0, 1]$; μ_k — наибольшее из чисел, удовлетворяющих неравенству

$$f_0(x^k - \mu_k e^{i_k}) - f_0(x^k) \leqslant -\frac{1}{2} \mu_k \left| \frac{\partial}{\partial x_{i_k}} f_0(x^k) \right|.$$

Для вычисления μ_k могут быть использованы следующие неравенства:

$$\mu_k \geqslant \min \left\{ x_{i_k}^k - \alpha_{i_k}, \ \frac{\partial}{\partial x_{i_k}} f_0(x^k)/\gamma \right\} \quad \text{при} \quad \frac{\partial}{\partial x_{i_k}} f_0(x^k) \geqslant 0;$$

$$\mu_k \geqslant -\min \left\{ \beta_{i_k} - x_{i_k}^k, \ \left| \frac{\partial}{\partial x_{i_k}} f_0(x^k) \right| / \gamma \right\} \quad \text{при} \quad \frac{\partial}{\partial x_{i_k}} f_0(x^k) \leqslant 0.$$

Библиографические указания. Пункт 1 основан на результатах работы [161], а пункт 2 — на результатах работы [131].

5.24. Релаксационные методы для общих задач нелинейного программирования

Задача 0. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0: R^n \to R^1$ и заданного множества

$$X = \{x \mid g_i(x) = 0, \quad i = 1, \ldots, m_1; \quad g_i(x) \leqslant 0, \quad i = m_1 + 1, \ldots, m; \quad x_j \geqslant 0, \quad j = 1, \ldots, n; \quad x \in R^n\}.$$

Предположения 0. (*i*) — функции $f_0(x)$, $g_i(x)$, $i = 1, \ldots, m$ — непрерывно дифференцируемые.

Определение 0. Точки $x \in X^0$, где

$$X^0 = \{x \mid g_i(x) = 0, \quad i = 1, \ldots, m_1; \quad g_i(x) < 0, \quad i = m_1 + 1, \ldots, m; \quad x_j > 0, \quad j = 1, \ldots, n; \quad x \in R^n\},$$

называют внутренними точками множества X, а точки $x \in X \setminus X^0$ — граничными.

В приводимых ниже методах решение задачи 0 сводится к нахождению предельных точек (при $t \to \infty$) решения $x(x^0, t)$ задачи Коши для системы

$$\frac{dx}{dt} = -M(x)\nabla f_0(x), \quad x(0) = x^0 \in X^0,$$

где $M(x)$ — матрица размера $n \times n$, определяемая функциями ограничений задачи 0.

1. Непрерывный вариант

Алгоритм 1

I. Выбрать произвольное начальное приближение $x = x^0 \in X^0$.

II. Определить на полупрямой $w \geqslant 0$ непрерывную функцию $\xi(w)$ скалярного аргумента такую, что

$$\xi(w) > 0 \quad \text{при} \quad w > 0 \quad \text{и} \quad \xi(0) = 0$$

(обычно принимают $\xi(w) = w$).

III. Вычислить $n \times m$-матрицу $g_x(x)$ (у которой (i, j)-й элемент есть $\partial g_j(x)/\partial x_i$).

IV. Вычислить диагональную $n \times n$-матрицу $\mathscr{D}(\xi^n(x))$ (у которой j-е, $j = 1, \ldots, n$, диагональные элементы есть $\xi(x_j)$).

Если ограничения $x_j \geqslant 0$, $j = 1, \ldots, n$, в задаче 0 отсутствуют, то следует положить $\mathscr{D}(\xi^n(x)) = I_n$, где I_n — единичная $n \times n$-матрица.

V. Вычислить диагональную $m \times m$-матрицу $\mathscr{D}(\xi^m(-g(x)))$ (у которой i-е, $i = 1, \ldots, m$, диагональные элементы есть $\xi(-g_i(x))$).

VI. Вычислить $n \times 1$-матрицу $\nabla_x f_0(x)$.

VII. Вычислить $n \times n$-матрицу

$$M(x) = \mathscr{D}(\xi^n(x))[I_n - [(g_x(x))^T \mathscr{D}(\xi^n(x)) g_x(x) + \mathscr{D}(\xi^m(-g(x)))]^{-1} (g_x(x))^T \mathscr{D}(\xi^n(x))], \tag{5.87}$$

где I_n — единичная $n \times n$-матрица; индекс «T» обозначает транспонирование.

VIII. Найти решение $x(x_0, t)$ задачи Коши для системы

$$\dot{x} = -M(x)\nabla_x f_0(x), \quad x(0) = x^0. \qquad (5.88)$$

Теорема 1. *Пусть: (i) — на компактном множестве X функции $f_0(x,)$ $g_i(x)$, $i = 1, \ldots, m$ — непрерывно дифференцируемы; (ii) — на множестве X выполнены условия регулярности ограничений*

$$g_i(x) \leqslant 0, \quad i = 1, \ldots, m; \quad x_j \geqslant 0, \quad j = 1, \ldots, n;$$

(ограничения $g_i(x) \leqslant 0$, $i = 1, \ldots, m$, и $x_j \geqslant 0$, $j = 1, \ldots, n$, удовлетворяют условию регулярности в точке x, если $g_i(x)$, $i \in \overline{1, m}$ — непрерывно дифференцируемы, а векторы e^j, $j \in \{j \mid x_j = 0, j \in \overline{1, n}\}$ и $\nabla g_i(x)$, $i \in \{i \mid g_i(x) = 0, i \in \overline{1, m}\}$ — линейно-независимы); *(iii) — все стационарные точки из множества X изолированы; (iv) — для каждого $x^0 \in X$ система (5.88) определяет единственное решение $x(x^0, t)$. Тогда для любых нестационарных точек $x^0 \in X^0$ решение $x(x^0, t)$ системы (5.88) сходится (при $t \to \infty$) к допустимой стационарной точке, в которой выполнены необходимые (в случае задач выпуклого программирования — и достаточные) условия минимума задачи 0.*

Теорема 1'. *Пусть: (i) — $f_0(x)$ — выпуклая, непрерывно дифференцируемая функция; (ii) — $g_i(x)$, $i = 1, \ldots, m$ — линейные функции; (iii) — всюду на компактном множестве X выполнены условия регулярности ограничений*

$$g_i(x) \leqslant 0, \quad i = 1, \ldots, m; \quad x_j \geqslant 0, \quad j = 1, \ldots, n.$$

Тогда решения системы (5.88) (при $t \to \infty$) сходятся к множеству решений задачи 0 при любых $x^0 \in X^0$.

2. Дискретный вариант

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0 \in X^0$.

II. Положить $k = 0$

Основной цикл. III. Вычислить $\nabla f_0(x^k)$.

IV. Вычислить по (5.87) $n \times n$-матрицу $M(x^k)$.

V. Вычислить значение шагового множителя ρ_k, удовлетворяющее условиям теоремы 2.

VI. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k M(x^k)\nabla f_0(x^k).$$

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполнены условия теоремы 1 и условия (i) —*

$$\|\nabla f_0(\bar{x}) - \nabla f_0(\bar{y})\| \leqslant \gamma \|\bar{x} - \bar{y}\|, \quad 0 < \gamma < \infty, \quad \forall \bar{x}, \bar{y} \in X;$$

(ii) — функции $g_i(x)$, $i = 1, \ldots, m$ — линейные; (iii) — шаговые множители ρ_k таковы, что

$$0 < \rho_k < \min\{1/\mu, 1/\nu, 2/\lambda\gamma\},$$

где

$$\lambda = \max_{x \in X} \max_{y \in R^n} \frac{y^T M(x) y}{\|y\|^2} < \infty;$$

$$\mu = \max_{j \in [1:n]} \max_{x \in X} \frac{\partial H(x, v)}{\partial x_j}; \quad \nu = \max_{i \in [1:m]} \max_{x \in X} v_i(x);$$

$$H(x, v) = f_0(x) + \sum_{i=1}^{m} v_i(x) g_i(x);$$

вектор $v(x) \triangleq (v_1(x), \ldots, v_m(x))$ является решением системы линейных уравнений (все матрицы определяются в алгоритме 1)

$$[(g_x(x))^T \mathcal{D}(\xi^n(x)) g_x(x) + \mathcal{D}(\xi^m(-g_x(x)))] v(x) +$$
$$+ (g_x(x))^T \mathcal{D}(\xi^n(x)) \nabla f_0(x) = 0;$$

(iv) — скалярная функция $\xi(w)$ *выбрана в виде* $\xi(w) = w$.

Тогда при любых нестационарных начальных точках $x^0 \in X^0$ *последовательность* $\{x^k\}_{k=0}^{\infty}$, *порожденная алгоритмом 2, сходится к допустимой стационарной точке, в которой выполнены необходимые (в случае задач выпуклого программирования и достаточные) условия минимума задачи 0, причем* $x^k \in X^0$, $f_0(x^{k+1}) \leqslant f_0(x^k)$ *для* $k = 0, 1, \ldots$. *Если, кроме того,*

$$\beta_1 \|z\|^2 \leqslant z^T H_{zz}(x, v(x)) z \leqslant \beta_2 \|z\|^2$$

при всех $x \in X$, $z \in R^{m+n-m_1}$,

где $(m + n - m_1) \times (m + n - m_1)$*-матрица* $H_{zz}(x, v(x))$, *определяется по формуле*

$$H_{zz}(x, v(x)) = \begin{pmatrix} \frac{\partial^2 H(x, v(x))}{\partial x^2} & 0 \\ 0 & \mathcal{D}(\tilde{v}(x)) \end{pmatrix};$$

$\tilde{v}(x) = (v^{m_1+1}(x), \ldots, v^m(x))$, $\mathcal{D}(\tilde{v}(x))$ — *диагональная матрица, элементы которой являются соответствующими координатами вектора* $\tilde{v}(x)$, *то для любой начальной точки* $x^0 \in X^0$ *справедливо*

$$\left\| \frac{\partial H(x^k, v(x^k))}{\partial x} \right\|^2 + \|\mathcal{D}((-g(x^k))^{1/2} v(x^k)\|^2 \leqslant$$
$$\leqslant \left[\left\| \frac{\partial H(x^0, v(x^0))}{\partial x} \right\|^2 + \|\mathcal{D}((-g(x^0))^{1/2} v(x^0)\|^2 \right] [1 - \rho_k \beta_1 + \rho_k^2 \beta_2^2]^k$$

и последовательность $\{x^k\}_{k=0}^{\infty}$ *сходится к единственной точке минимума* x^* *задачи 0.*

Здесь через $\mathcal{D}((a)^{1/2})$ обозначено диагональную $m \times m$-матрицу, у которой диагональные элементы равны $(a_i)^{1/2}$, $i = 1, \ldots, m$, $a = (a_1, \ldots, a_m)$.

Библиографические указания. При написании параграфа была использована работа [139]. Другие сведения о сходимости методов, оценках скорости сходимости, рекомендации по практическому использованию методов, результаты экспериментального апробирования методов можно найти в работах [134, 136, 137, 138, 139].

5.25. Методы фейеровских приближений решения задач выпуклого программирования с негладкими ограничениями

1. Общий случай

Определение 1. Отображение $\psi : R^n \to 2^{R^n}$ называется фейеровским, если

$$X(\psi) \triangleq \{x \mid x \in \psi(x)\} \neq \varnothing;$$

$$\|z - y\| < \|x - y\|, \quad \forall z \in \psi(x), \quad \forall y \in X(\psi), \quad \forall x \in X(\psi).$$

Класс фейеровских отображений обозначим через Φ и выделим подкласс Φ_0 класса Φ условием

$$\Phi_0 = \{\psi \mid x \in \psi(x) \Rightarrow \psi(x) = x; \quad \psi \in \Phi\}.$$

Определение 1'. Отображение $\psi : R^n \to 2^{R^n}$ называется замкнутым, если из соотношений

$$\lim_{k \to \infty} x^k = x';$$

$$y^k = \psi(x^k), \quad k = 1, 2, \ldots;$$

$$\lim_{k \to \infty} y^k = y'$$

следует $y' \in \psi(x')$.

Задача 1. Найти $\arg\max\limits_{x \in X(\psi)} (c, x)$ для заданного вектора $c \in R^n$ и заданного фейеровского отображения $\psi \in \Phi_0$.

Алгоритм 1

Начало. I. Задать фейеровское отображение $\psi \in \Phi_0$.

II. Выбрать произвольное начальное приближение $x^0 \in R^n$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям теоремы 1.

V. Вычислить следующее приближение

$$x^{k+1} \in \psi(x^k + \rho_k c).$$

VI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Пусть выполняются условия: (**i**) — фейеровское отображение $\psi \in \Phi_0$ — замкнуто; (**ii**) — множество $X \triangleq X(\psi)$ — ограничено; (**iii**) — для некоторого $\delta_0 > 0$ существует $\varepsilon_0 > 0$ такое, что*

$$\min_{z \in X} \|x - z\| - \min_{z \in X} \|y - z\| \geqslant \varepsilon_0, \quad \forall x \in X_{\delta_0}, \quad \forall y \in \psi(x);$$

(**iv**) —

$$\rho_k > 0, \quad k = 0, 1, \ldots; \ \lim_{k \to \infty} \rho_k = 0; \quad \sum_{k=0}^{\infty} \rho_k = \infty.$$

Тогда предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, принадлежат множеству решений задачи 1.

Здесь X_δ обозначает δ-окрестность множества X.

Замечание 1. Если для задачи

$$\text{найти } \arg\max_{x\in X}(c,\ x),\quad c\in \boldsymbol{R}^n, \tag{5.89}$$

$$X=\{x\,|\,f_j(x)\leqslant 0,\quad j=1,\ \ldots,\ m,\quad x\in \boldsymbol{R}^n\} \tag{5.90}$$

с выпуклыми функциями $f_j(x)$, $j=1, \ldots, m$, и ограниченным множеством X фейеровское отображение ψ определить по правилу

$$\psi(x)\triangleq\{x-\lambda\mu(x,\ h)\,|\,h\in G(x)\}, \tag{5.91}$$

где параметр $\lambda\in(0,\ 2)$;

$$\mu(x,\ h)=\begin{cases}0, & \text{если } f(x)\leqslant 0,\\ (f(x)/\|h\|^2)\,h, & \text{если } f(x)>0;\end{cases}$$

$$f(x)=\max_{1\leqslant j\leqslant m} f_j(x)\quad \text{или}\quad f(x)=\sum_{j=1}^{m}\max\{0,\ f_j(x)\};$$

$G(x)$ — множество обобщенных градиентов выпуклой функции $f(\cdot)$ в точке x, то условия (*i*), (*ii*), (*iii*) теоремы 1 будут выполняться.

Таким образом, решение задачи (5.89) — (5.90) можно получить с помощью алгоритма 1, в котором фейеровское отображение ψ определяется по (5.91).

2. Случай кусочно линейных ограничений

Задача 2. Найти $\arg\min_{x\in X}(c,\ x)$ при ограничениях

$$X=\left\{x\,\middle|\,\sum_{i=1}^{t}\max_{l\in\mathcal{I}_i}[(c^{jl},\ x)-a_{jl}]-b_j\leqslant 0,\ j=1,\ldots,m,\ x\in\boldsymbol{R}^n\right\},$$

где $c\in\boldsymbol{R}^n$; $c^{jl}\in\boldsymbol{R}^n$ при $j\in[1:m]$, $l\in\bigcup_i\mathcal{I}_i$; a_{jl}, $j\in[1:m]$, $l\in\bigcup_i\mathcal{I}_i$; b_j, $j\in[1:m]$ — действительные числа; $t\geqslant 1$ — натуральное число; $\mathcal{I}_i$— множества индексов.

Предположение 2. Множество решений X^* задачи 2 непусто.

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $x^0\in\boldsymbol{R}^n$.

II. Выбрать произвольный параметр $\lambda\in(0,\ 2)$ и произвольные константы $\alpha_j>0$, $j=1, \ldots, m$.

III. Положить $k=0$.

Основной цикл. IV. При каждом $j\in[1:m]$ вычислить индексы

$$r_{ji}=\min\{s\,|\max_{l\in\mathcal{I}_i}[(c^{jl},\ x^k)-a_{jl}]=(c^{js},\ x^k)-a_{js},\quad s\in\mathcal{I}_i\},$$

$$i=1,\ \ldots,\ t,\quad j=1,\ \ldots,\ m.$$

V. При каждом $j\in[1:m]$ вычислить n-мерный вектор

$$h^j(x^k)=\sum_{i=1}^{t}c^{jr_{ji}},\quad j=1,\ \ldots,\ m.$$

VI. Найти X-разделяющую пару $(h(x^k), \beta(x^k))$ по правилам

$$h(x^k) = \sum_{j=1}^{m} \alpha_j h^j(x^k);$$

$$\beta(x^k) = \sum_{j=1}^{m} \alpha_j \left[\sum_{i=1}^{t} \max_{l \in \mathcal{I}_i} [(c^{il}, x^k) - a_{jl}] - b_j \right]_+.$$

Пара $(h(x^k), \beta(x^k))$ является X-разделяющей, если для любого фиксированного $x^k \notin X$ полупространство, отвечающее неравенству

$$(h(x^k), x - x^k) + \beta(x^k) \leqslant 0,$$

содержит множество X. X-разделяющую пару $(h(x^k), \beta(x^k))$ можно также вычислить по формулам

$$\beta(x^k) = \max_{j \in [1:m]} \left[\sum_{i=1}^{t} \max_{l \in \mathcal{I}_i} [(c^{il}, x^k) - a_{jl}] - b_j \right]_+;$$

$$h(x^k) = h^{j_k}(x^k),$$

$$j_k = \arg \max_{j \in [1:m]} \left[\sum_{i=1}^{t} \max_{l \in \mathcal{I}_i} [(c^{il}, x^k) - a_{jl}] - b_j \right]_+, \quad x^k \notin X,$$

где $[\sigma]_+ = \max\{0, \sigma\}$.

VII. Вычислить значение ρ_k, удовлетворяющее условиям теоремы 2.

VIII. Вычислить вектор

$$\bar{x}^{k+1} = \begin{cases} x^k - \lambda \dfrac{\beta(x^k)}{\|h(x^k)\|^2} h(x^k), & \beta(x^k) > 0, \\ x^k, & \beta(x^k) \leqslant 0. \end{cases}$$

IX. Вычислить следующее приближение

$$x^{k+1} = \bar{x}^{k+1} - \rho_k c.$$

X. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Если выполняется предположение 2 и последовательность $\{\rho_k\}_{k=0}^{\infty}$ удовлетворяет условиям*

$$\rho_k > 0; \quad \rho_k \to 0 \quad \textit{при} \quad k \to \infty; \quad \sum_{k=0}^{\infty} \rho_k = \infty,$$

то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, сходится к элементу $x^* \in X^*$.

Библиографические указания. Пункт 1 написан на основании работы [145], при написании пункта 2 использовались работы [140, 142, 143, 144, 146].

5.26. Двойственные методы

1. Приближенный двойственный метод решения задач выпуклого программирования

Задача 1. Найти $\arg\max_{x \in X} f_0(x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X = \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m, \quad x \in Q\},$$

где Q — выпуклое множество в R^n; $f_i : R^n \to R^1$, $i = 1, \ldots, m$.

Предположения 1. (i) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — выпуклые вверх на выпуклом множестве Q.

Двойственной к задаче 1 является следующая задача 1′: найти $\arg\min_{y \geqslant 0} g_0(y)$, где $g_0(y) \triangleq \sup_{x \in Q} \varphi(x, y)$, $\varphi(x, y)$ — функция Лагранжа задачи 1.

В приводимом ниже методе вводится модифицированная функция Лагранжа $\psi(x, y)$ в виде

$$\psi(x, y) = \sup_{z \geqslant 0} \left[f_0(x) + \sum_{i=1}^{m} (f_i(x) - z_i) y_i - \sum_{i=1}^{m} r_i (f_i(x) - z_i) \right], \quad (5.92)$$

где $y = (y_1, \ldots, y_m)$, $z = (z_1, \ldots, z_m)$, $r_i(\xi)$ — дважды непрерывно дифференцируемая функция вещественной переменной ξ такова, что

$$r_i(0) = \frac{dr_i(0)}{d\xi} = 0; \quad \frac{d^2 r_i(0)}{d\xi^2} \geqslant \gamma > 0.$$

На k-й итерации алгоритма вычисляется точка x^k, являющаяся приближенным решением некоторой вспомогательной задачи условной максимизации (точность решения вспомогательной задачи задается некоторой управляющей последовательностью), а точка y^{k+1} вычисляется в направлении антиградиента модифицированной функции Лагранжа, вычисленного в точке (x^k, y^k).

При определенных условиях получаемая последовательность двойственных переменных $\{y_k\}_{k=0}^{\infty}$ будет сходиться к непустому множеству Y^* решений двойственной задачи, а $\{x^k\}_{k=0}^{\infty}$ будет являться обобщенным решением задачи 1, т. е.

$$\underline{\lim}_{k \to \infty} f_i(x^k) \geqslant 0, \quad i = 1, \ldots, m, \quad \lim_{k \to \infty} f_0(x^k) = \sup_{x \in X} f_0(x).$$

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение двойственной переменной $y^0 \in R^m$.

II. Выбрать дважды непрерывно дифференцируемые функции $r_i(\xi)$, $i = 1, \ldots, m$, вещественной переменной ξ такие, что

$$r_i(0) = \frac{dr_i(0)}{d\xi} = 0, \quad \frac{d^2 r_i(\xi)}{d\xi^2} \geqslant \gamma > 0, \quad i = 1, \ldots, m. \quad (5.93)$$

III. Определить модифицированную функцию Лагранжа

$$\psi(x,\ y) = \max_{z \geqslant 0}\left[f_0(x) + \sum_{i=1}^{m}(f_i(x) - z_i)\,y_i - \sum_{i=1}^{m} r_i(f_i(x) - z_i)\right],$$

где $z = (z_1,\ \ldots,\ z_m)$; $y = (y_1,\ \ldots,\ y_m)$.

IV. Положить $k = 0$.

Основной цикл. V. Найти шаговый множитель ρ_k и элемент δ_k управляющей последовательности, удовлетворяющие условиям теоремы 3.

VI. Вычислить точку $x^k \in Q$, удовлетворяющую условию

$$\psi(x^k,\ y^k) \geqslant \sup_{x \in Q} \psi(x,\ y^k) - \delta_k.$$

VII. Вычислить вектор $\nabla_y \psi(x^k,\ y^k)$ (градиент функции $\psi(x,\ y)$ по переменной y в точке $(x^k,\ y^k)$) по правилу

$$\frac{\partial \psi(x^k,\ y^k)}{\partial y_i} = \min\{f_i(x^k),\ t_i(y_i^k)\},\quad i = 1,\ \ldots,\ m,$$

где $t_i(\eta)$, $i = 1,\ \ldots,\ m$ — функция, обратная функции $\dfrac{dr_i(\xi)}{d\xi}$.

VIII. Вычислить следующее приближение

$$y^{k+1} = y^k - \rho_k \nabla_y \psi(x^k,\ y^k).$$

IX. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Пусть выполняется предположение 1 и пусть (**ii**) — множество Y^* решений двойственной задачи 1 непусто; (**iii**) — управляющая последовательность $\{\delta_k\}_{k=0}^{\infty}$ такова, что*

$$\delta_k > 0,\quad k = 0,\ 1,\ \ldots,\quad \sum_{k=0}^{\infty}(\delta_k)^{1/2} < \infty;$$

*(**iv**) — последовательность шаговых множителей $\{\rho_k\}_{k=0}^{\infty}$ удовлетворяет условию*

$$0 < \bar{\rho} \leqslant \rho_k \leqslant \hat{\rho} < 2\gamma,\quad k = 0,\ 1,\ \ldots,$$

где γ — число, фигурирующее в неравенствах (5.93).

Тогда при произвольном начальном приближении $y^0 \in R^m$ последовательности $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$, порожденные алгоритмом 1, обладают свойствами

$$\lim_{k \to \infty} y^k = y^* \in Y^*,$$

$\{x^k\}_{k=0}^{\infty}$ — решение обобщенной задачи, соответствующей задаче 1, т. е.

$$\varliminf_{k \to \infty} f_i(x^k) \geqslant 0,\quad i = 1,\ \ldots,\ m,\quad \lim_{k \to \infty} f_0(x^k) = \bar{v},$$

где $\bar{v}$ — экстремальное значение обобщенной задачи.

Замечание 1. Если дополнительно выполняется соотношение двойственности, т. е. $v = \breve{v}$, где $\breve{v}$ — экстремальное значение двойственной задачи 1′, то последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, является обобщенным решением задачи 1, т. е.

$$\varliminf_{k\to\infty} f_i(x^k) \geqslant 0, \quad i = 1, \ldots, m; \quad \lim_{k\to\infty} f_0(x^k) = v.$$

Здесь v — экстремальное значение задачи 1.

Определение обобщенной задачи. Последовательность $\{x^k\}_{k=0}^{\infty}$, $x^k \in Q$ (обозначим ее через w) является обобщенным решением задачи 1, если:

а) существует конечный или бесконечный предел

$$\lim_{k\to\infty} f_0(x^k)$$

(обозначим этот предел через $f_0(w)$);

б)

$$\varliminf_{k\to\infty} f_i(x^k) \geqslant 0, \quad i = 1, \ldots, m.$$

Множество обобщенных решений задачи 1 обозначим через W. Задачу максимизации функции $f_0(w)$ на множестве обобщенных решений W называют обобщенной задачей 1.

Экстремальное значение $\bar{v}$ обобщенной задачи 1 определяется по формуле

$$\bar{v} = \begin{cases} \sup\limits_{w\in W} f_0(w), & \text{если} \quad W \neq \varnothing, \\ -\infty, & \text{если} \quad W = \varnothing. \end{cases}$$

Глубже с понятием и свойствами обобщенной задачи можно ознакомиться в работах [80, 81].

Теорема 1′. *Пусть выполняется предположение 1, условия* (*ii*), (*iv*) *теоремы 1 и пусть, кроме того,* (*v*) — *множество* Y^* — *ограничено;* (*vi*) — *управляющая последовательность* $\{\delta_k\}_{k=0}^{\infty}$ *такова, что*

$$\delta_k > 0, \quad k = 0, 1, \ldots, \quad \lim_{k\to\infty} \delta_k = 0.$$

Тогда для произвольного $y^0 \in R^m$ *последовательности* $\{x^k\}_{k=0}^{\infty}$, $\{y^k\}_{k=0}^{\infty}$, *порожденные алгоритмом 1, таковы, что*

$$\min_{y^*\in Y^*} \| y^k - y^* \| \to 0, \quad k \to \infty;$$

$$\varliminf_{k\to\infty} f_i(x^k) \geqslant 0, \quad i = 1, \ldots, m;$$

$$\lim_{k\to\infty} f_0(x^k) = v,$$

где v — *экстремальное значение задачи 1.*

Для выполнения условий (*ii*), (*v*) теоремы достаточно, чтобы выполнялось условие Слейтера для задачи 1.

Замечания. В работе [549] частичные случаи приведенных выше теорем получены для модифицированной функции Лагранжа при

$$r_i(\xi) = (\gamma/2)\xi^2.$$

2. Двойственный алгоритм переменной метрики

Задача 2. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и множества

$$X = \{x \mid g_j(x) \leqslant 0, \quad j = 1, \ldots, m, \quad f_i(x) = 0, \quad i = 1, \ldots, l, \quad x \in \boldsymbol{R}^n\},$$

где $g_j : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $j = 1, \ldots, m$; $f_i : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $i = 1, \ldots, l$.

В дальнейшем будем обозначать $g(x) = (g_1(x), \ldots, g_m(x))$, $f(x) = (f_1(x), \ldots, f_l(x))$.

Предположение 2. (*i*) — существует тройка Куна — Таккера $z^* = (x^*, u^*, v^*)$ задачи 2; (*ii*) — функции $f_i(x)$, $i = 1, \ldots, l$ и $g_j(x)$, $j = 1, \ldots, m$, имеют непрерывные по Липшицу вторые производные в окрестности точки x^*.

Определение 2. Тройка Куна — Таккера $z^* = (x^*, u^*, v^*)$ задачи 2 удовлетворяет условиям единственности якобиана, если выполняются одновременно следующие условия:

а) $u_j^* > 0$, если $j \in \mathcal{I} \triangleq \{j \mid g_j(x^*) = 0\}$;

б) векторы $\nabla g_j(x^*)$, $j \in \mathcal{I}$, $\nabla f_i(x^*)$, $i = 1, \ldots, l$ — линейно-независимы;

в) для каждого ненулевого вектора y, удовлетворяющего равенствам

$$y^T \nabla g_j(x^*) = 0, \quad \forall j \in \mathcal{I};$$
$$y^T \nabla f_i(x^*) = 0, \quad i = 1, \ldots, l,$$

выполнено

$$y^T \nabla_{xx}^2 \varphi(z^*) y > 0,$$

где $\nabla_{xx}^2 \varphi(z^*)$ — матрица вторых производных по x функции Лагранжа

$$\varphi(z) = f_0(x) + u^T g(x) + v^T f(x), \quad z = (x, u, v),$$

вычисленная в точке $z^* = (x^*, u^*, v^*)$.

Алгоритм 2

Начало. I. Выбрать начальное приближение $z^0 = (x^0, u^0, v^0) \in R^{n+m+l}$ для тройки Куна — Таккера $z^* = (x^*, u^*, v^*)$ задачи 2.

II. Выбрать начальное приближение A_0 для матрицы, обратной к гессиану функции Лагранжа задачи 2, вычисленному в точке z^*.

III. Положить $k = 0$.

Основной цикл. IV. Найти ближайшую точку Куна — Таккера (u^{k+1}, v^{k+1}) для задачи квадратичного программирования

$$\arg\min_{(u,v)} \left[\frac{1}{2}(\nabla f_0(x^k) + \nabla g(x^k) u + \nabla f(x^k) v)^T A_k (\nabla f_0(x^k) + \right.$$
$$\left. + \nabla g(x^k) u + \nabla f(x^k) v) - u^T g(x^k) - v^T f(x^k)\right] \qquad (5.94)$$

при ограничении $u \geqslant 0$.

V. Вычислить вектор

$$x^{k+1} = x^k - A_k [\nabla f_0(x^k) + \nabla g(x^k) u^{k+1} + \nabla f(x^k) v^{k+1}]$$

и положить $z^{k+1} = (x^{k+1}, u^{k+1}, v^{k+1})$.

VI. Вычислить матрицу A_{k+1}, положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Пусть выполняются предположения 2 и (**iii**) — тройка Куна — Таккера $z^* = (x^*, u^*, v^*)$ удовлетворяет условию единственности якобиана; (**iv**) — матрица $\nabla^2_{xx}\varphi(z^*)$ вторых производных по x функции Лагранжа задачи 2 неособенная.*

Тогда для любого $\alpha \in (0, 1)$ существуют два положительных числа $\varepsilon(\alpha)$ и $\delta(\alpha)$ таких, что если

$$\|(x^k, u^k, v^k) - z^*\| \leqslant \varepsilon(\alpha)$$

и A_k — симметричная $n \times n$-матрица, удовлетворяющая неравенству

$$\|A_k - [\nabla^2_{xx}\varphi(z^*)]^{-1}\| \leqslant \delta(\alpha),$$

то существует решение (u^{k+1}, v^{k+1}) задачи (5.94) такое, что ближайшая к (x^k, u^k, v^k) точка $(x^{k+1}, u^{k+1}, v^{k+1})$ удовлетворяет неравенству

$$\|(x^{k+1}, u^{k+1}, v^{k+1}) - z^*\| \leqslant \alpha \|(x^k, u^k, v^k) - z^*\|.$$

Теорема 2'. *Пусть выполнены все предположения теоремы 2 и пусть $\{s_k\}_{k=0}^{\infty}$ — последовательность положительных чисел, удовлетворяющая условию $s_k \leqslant k$, $k = 0, 1, \ldots$, $s_{k+1} \geqslant s_k$.*

Если в алгоритме 2 z^0 достаточно близко к z^* и

$$\|A_k - [\nabla^2_{xx}\varphi(z^{s_k})]^{-1}\| \leqslant \beta_k,$$

где $\{\beta_k\}_{k=0}^{\infty}$ — последовательность неотрицательных чисел, ограниченная достаточно малым числом, то последовательность $\{z^k\}_{k=0}^{\infty}$, порождаемая алгоритмом 2, существует и сходится к z^* по крайней мере с линейной скоростью. Если, кроме того, $s_k \to \infty$ и $\beta_k \to 0$ при $k \to \infty$, то $\{z^k\}_{k=0}^{\infty}$ сходится к z^* по крайней мере сверхлинейно.

Теорема 2''. *Пусть выполняются предположения 2 и начальная точка z^0 и начальная матрица A_0 достаточно близки к z^* и $\nabla^2_{xx}\varphi(z^*)$ соответственно. Тогда, если в алгоритме 2 матрицы*

A_{k+1}, $k = 0, 1, \ldots$, *вычислять по рекуррентной формуле*

$$A_{k+1} = A_k + \frac{(h - A_k y) h^T + h (h - A_k y)^T}{h^T y} - \frac{y^T (h - A_k y) h h^T}{(h^T y)^2},$$

$$k = 0, 1, \ldots,$$

где

$$h = x^{k+1} - x^k; \; y = \nabla_x \varphi (x^{k+1}, u^{k+1}, v^{k+1}) - \nabla_x \varphi (x^k, u^{k+1}, v^{k+1}),$$

то последовательность $\{z^k\}_{k=0}^{\infty}$, *генерируемая алгоритмом 2, существует и сходится сверхлинейно к* z^*

Замечание 2. В [490] отмечается, что локальная сверхлинейная сходимость может быть достигнута также в случае, когда матрицы A_{k+1} вычислять по одной из следующих формул:

$$A_{k+1} = A_k - \frac{A_k y y^T A_k}{y^T A_k y} + \frac{h h^T}{h^T y};$$

$$A_{k+1} = A_k + \frac{(h - A_k y) h^T + h (h - A_k y)^T}{2 h^T y}; \quad (5.95)$$

$$A_{k+1} = A_k + \frac{(h - A_k y) y^T + y (h - A_k y)^T}{2 y^T y}.$$

Если дополнительно предполагать, что матрица $\nabla^2_{xx} \varphi (z^*)$ положительно определенная, то теорема 2″ справедлива для алгоритма 2, в котором матрицы A_{k+1}, $k = 0, 1, \ldots$, вычисляются по следующей рекуррентной формуле

$$A_{k+1} = A_k + \frac{(h - A_k y) y^T + y (h - A_k y)^T}{y^T y} - \frac{y^T (h - A_k y) y y^T}{(y^T y)^2},$$

$$k = 0, 1, \ldots .$$

Библиографические указания. Пункт 1 основан на результатах работы [87], при написании пункта 2 использовались работы [488, 490].

5.27. Глобально сходящийся метод

Задача 1. Найти $\arg\min_{x \in X} f_0 (x)$ для заданной функции $f_0 : R^n \to R^1$ и множества

$$X = \{x \mid f_j (x) \leqslant 0, \quad j = 1, \ldots, m, \quad x \in R^n\},$$

где $f_j : R^n \to R^1$, $j = 1, \ldots, m$.

Предположения 1. (*i*) — функция $f_0 (x)$ ограничена снизу в R^n; (*ii*) — функции $f_i (x)$, $i = 0, 1, \ldots, m$ — непрерывно дифференцируемы в R^n; (*iii*) — функции $f_j (x)$, $j = 1, \ldots, m$ — выпуклы.

В приводимом здесь методе на k-й итерации для определения направления движения к следующему приближению x^{k+1} решается

некоторая вспомогательная задача квадратичного программирования, а для определения шагового множителя решается (приближенно) задача одномерной условной минимизации.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать произвольную последовательность $\{\varepsilon_k\}_{k=0}^{\infty}$ такую, что $\varepsilon_k \geqslant 0$, $k = 0, 1, \ldots, \sum_{k=0}^{\infty} \varepsilon_k < \infty$.

III. Выбрать произвольные константы $\alpha > 0$ и $\delta > 0$.

IV. Определить функцию

$$g_\alpha(x) = f_0(x) + \alpha \sum_{j=1}^{m} [f_j(x)]_+,$$

где

$$[\sigma]_+ = \max\{0, \sigma\}.$$

V. Положить $k = 0$.

Основной цикл. VI. Найти матрицу H_k, удовлетворяющую условию (*vi*) теоремы 1.

VII. Вычислить решение h^k следующей задачи квадратичного программирования:

$$\text{найти } \arg\min_h \left[(\nabla f_0(x^k), h) + \frac{1}{2}(H_k h, h)\right]$$

при ограничениях

$$f_j(x^k) + (\nabla f_j(x^k), h) \leqslant 0, \quad j = 1, \ldots, m.$$

VIII. Вычислить шаговый множитель $\rho_k \in [0, \delta]$, удовлетворяющий неравенству

$$g_\alpha(x^k + \rho_k h^k) \leqslant \min_{0 \leqslant \rho \leqslant \delta} g_\alpha(x^k + \rho h^k) + \varepsilon_k.$$

IX. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k.$$

X. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 1. *Пусть выполняются предположения 1 и* (***iv***) *— множество X компактно;* (*v*) *— множество*

$$X^0 \triangleq \{x \mid f_j(x) < 0, \quad j = 1, \ldots, m, \quad x \in R^n\}$$

— непусто; (***vi***) *— существуют положительные числа β_1 и β_2 такие, что для каждого $k \geqslant 0$ и для любого $x \in R^n$ выполняется*

$$\beta_1 \|x\|^2 \leqslant (H_k x, x) \leqslant \beta_2 \|x\|^2.$$

Тогда для любого начального приближения $x^0 \in R^n$ существует положительное число $\bar{\alpha}$ такое, что если в алгоритме 1 константу α выбрать из условия

$$\alpha \geqslant \max\{\bar{\alpha}, 1\},$$

то предельные точки последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, являются точками Куна — Таккера задачи 1.*

Библиографические указания. Параграф написан на основании работы [489].

5.28. Стохастические квазиградиентные методы

Задача 0. Найти $\arg\min_{x \in X} f_0$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного множества $X \subset \boldsymbol{R}^n$.

Предположения 0. (*i*) — функция f_0 выпукла вниз; (*ii*) — множество X выпукло и замкнуто.

Стохастические квазиградиентные методы применяются для решения задач оптимизации, в которых нет достаточно полной информации о функциях цели, функциях ограничений и их производных. Эти методы основаны на понятии стохастического квазиградиента — случайного вектора ξ^k, удовлетворяющего равенству

$$E(\xi^k/x^0, \ldots, x^k) = \alpha_k \hat{\nabla} f_0(x^k) + b^k,$$

где α_k — неотрицательная случайная величина, b^k — случайный вектор, измеримые относительно σ-подалгебры, индуцированной семейством случайных величин $(x^0, \ldots, x^k)$; $\hat{\nabla} f_0(x^k)$ — обобщенный градиент функции f_0 в точке x^k.

Приведем примеры вычисления стохастических квазиградиентов для функции f_0.

Пример 1. Пусть функция $f_0(x)$ имеет ограниченные вторые производные. Рассмотрим вектор $\beta = (\beta_1, \ldots, \beta_n)$ с независимыми и равномерно распределенными на $[-1, 1]$ компонентами. Тогда вектор стохастического квазиградиента в k-й итерации можно вычислять по формуле

$$\xi^k = \sum_{s=1}^{s_k} \frac{f_0(x^k + \Delta_k \beta^{k,s}) - f_0(x^k)}{\Delta_k} \beta^{k,s},$$

где $\beta^{k,s}$, $s = 1, \ldots, s_k$ — серия независимых наблюдений β в k-й итерации, причем $s_k \geqslant 1$; $\Delta_k > 0$. Если случайные величины s_k, Δ_k измеримы относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами $x^0, \ldots, x^k$, то условное математическое ожидание такого вектора равно

$$E(\xi^k/x^k) = \frac{s_k}{3} \nabla f_0(x^k) + v^k \Delta_k,$$

где v^k — некоторый случайный вектор, измеримый относительно $\mathcal{B}_k$, $\|v^k\| \leqslant \text{const}$.

Пример 2. Пусть функция $f_0(x)$ имеет вид

$$f_0(x) = \sum_{i=1}^{r} p_i f_i(x),$$

где $p_i \geqslant 0$, $\sum_{i=1}^{r} p_i = 1$, и пусть функции $f_i(x)$, $i = 1, \ldots, r$ — дважды непрерывно дифференцируемы. Введем случайную величину ν, которая принимает значения $1, 2, \ldots, r$ с вероятностями $p_1, p_2, \ldots, p_r$. За вектор стохастического квазиградиента в k-й итерации можно взять

$$\xi^k = \sum_{s=1}^{s_k} \frac{f_{\nu_k}(x^k + \Delta_k \beta^{k,s}) - f_{\nu_k}(x^k)}{\Delta_k} \beta^{k,s},$$

где ν_k — реализация случайной величины ν в k-й итерации, вектор $\beta^{k,s}$ и числа s_k, Δ_k такие, как и в примере 1. Условное математическое ожидание такого вектора

$$E(\xi^k / x^k) = \frac{s_k}{3} \nabla f_0(x^k) + v^k \Delta_k,$$

где v^k — некоторый случайный вектор, измеримый относительно $\mathcal{B}_k$, $\| v^k \| \leqslant \text{const}$.

Пример 3. Пусть функция $f_0(x)$ имеет вид

$$f_0(x) = \max_{y \in Y} f(x, y),$$

где $f(x, y)$ — выпуклая вниз по x, дважды непрерывно дифференцируемая по x при каждом y функция; Y — выпуклое компактное множество.

Обобщенным градиентом функции f_0 в точке x является вектор

$$\hat{\nabla} f_0(x) = \left(\frac{\partial f(x, y)}{\partial x_1}, \ldots, \frac{\partial f(x, y)}{\partial x_n} \right) \Bigg|_{y = y(x)},$$

где вектор $y(x)$ такой, что

$$f(x, y(x)) = \max_{y \in Y} f(x, y).$$

За вектор стохастического квазиградиента в k-й итерации можно взять

$$\xi^k = \sum_{s=1}^{s_k} \frac{f(x^k + \Delta_k \beta^{k,s}, y(x^k)) - f(x^k, y(x^k))}{\Delta_k} \beta^{k,s},$$

где вектор $\beta^{k,s}$ и числа Δ_k, s_k определяются как и в примере 1. Условное математическое ожидание такого вектора равно

$$E(\xi^k / x^k) = \frac{s_k}{3} \hat{\nabla} f_0(x^k) + v^k \Delta_k,$$

где v^k — некоторый случайный вектор, измеримый относительно $\mathcal{B}_k$, $\| v^k \| \leqslant \text{const}$.

Пример 4. Если $f_0(x) = EF(x, \omega)$ и выполнены требования, достаточные для дифференцирования под знаком математического

ожидания, то в качестве ξ^k можно брать $\nabla_x F(x^k, \omega)$, так как $E(\xi^k/x^k) = \nabla f_0(x^k)$, или

$$\xi^k = \sum_{s=1}^{s_k} \frac{F(x^k + \Delta_k \beta^{k,s}, \omega) - F(x^k, \omega)}{\Delta_k} \beta^{k,s},$$

где $\beta^{k,s}$ определены в примере 1.

1. Метод проектирования стохастических квазиградиентов

Алгоритм 1

Начало. I. Выбрать произвольную начальную точку $x^0 \in R^n$, для которой $E\|x^0\|^2 < \infty$, задать последовательность шаговых множителей $\{\rho_k\}_{k=0}^{\infty}$ и последовательность нормирующих множителей $\{\gamma_k\}_{k=0}^{\infty}$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить случайный вектор $\xi^k(\omega)$, условное математическое ожидание которого

$$E(\xi^k/x^0, \ldots, x^k) = \alpha_k \hat{\nabla} f_0(x^k) + b^k,$$

где α_k— неотрицательная случайная величина и $b^k = (b_1^k, \ldots, b_n^k)$ — случайный вектор, зависящие от последовательности $x^0, \ldots, x^k$, или, более точно, измеримые относительно σ-подалгебры $\mathcal{B}_k$, индуцированной семейством случайных величин $(x^0, \ldots, x^k)$; $\hat{\nabla} f_0(x^k)$ — обобщенный градиент функции $f_0(\cdot)$ в точке x^k.

IV. Вычислить вектор

$$x^{k+1} = \pi_X(x^k - \rho_k \gamma_k \xi^k(\omega)), \qquad (5.96)$$

где π_X— оператор проектирования на множество X.

V. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть имеют место предположения 0 и пусть, кроме того, для любого числа $w < \infty$ найдется такое число $c_w < \infty$, что*

$$E(\|\xi^k\|^2/x^0, \ldots, x^k) \leqslant \eta_k^2 \leqslant c_w \quad \text{при} \quad \|x^i\| \leqslant w, \quad i = 0, \ldots, k.$$

Кроме того, величины ρ_k, γ_k измеримы относительно $\mathcal{B}_k$ и такие, что

$$\gamma_k > 0, \quad \gamma_k(\eta_k + \tau_k\|x^k\|) \leqslant \text{const},$$

где

$$\tau_k = 1 \quad \text{при} \quad \|b^k\| \neq 0, \quad \tau_k = 0 \quad \text{при} \quad \|b^k\| = 0;$$

$$\rho_k \geqslant 0, \quad \sum_{k=0}^{\infty} \rho_k \alpha_k = \infty, \quad \sum_{k=0}^{\infty} E\rho_k\|b^k\| < \infty.$$

Тогда найдется подпоследовательность x^{k_i} последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, для которой

$$\lim_{i\to\infty} f_0(x^{k_i}) = f_0(x^*) \ \text{п. н.},$$

или

$$\lim_{k\to\infty} \min_{i\leqslant k} f_0(x^i) = f_0(x^*) \ \text{п. н.}$$

Теорема 1'. *Пусть в дополнение к условиям теоремы 1 множество X^* ограничено. Тогда*

$$\inf_{x^*\in X^*} E\|x^* - x^k\|^2 \to 0 \quad \text{при} \quad k\to\infty.$$

Для последовательности точек x^0, x^1, ..., порожденной алгоритмом 1, в котором на шаге IV процедура (5.96) заменена процедурой

$$x^{k+1} = \pi_X(x^k - \rho_k \xi^k(\omega)),$$

имеет место следующая теорема 1'', дающая оценку скорости сходимости.

Теорема 1''. *Пусть выполнены предположения 0 и пусть существуют такие постоянные $\bar\beta$ и λ, что*

$$f_0(x) \geqslant f_0(x^*) + \beta\|x^* - x\|^2$$

при $x\in X$, и с вероятностью 1

$$E(\|\xi^k\|^2/x^0, \ldots, x^k) \leqslant \lambda.$$

Кроме того, ρ_k детерминированные и

$$\alpha_k \geqslant \alpha > 0, \quad \|b^k\| \leqslant r_k, \quad r_k \leqslant \rho_k, \quad r_k < r < 2\bar\beta\alpha.$$

Тогда существует такое число $c_k \leqslant$ const, что при $\rho_k = c_k/k$ последовательность $\{x^k\}_{k=0}^{\infty}$ с вероятностью 1 сходится к единственному решению x^*, причем

$$E\|x^* - x^k\|^2 = O(1/k).$$

Определение 1. Пусть G — некоторое подмножество пространства R^n. Последовательность $z^s(\omega)$, $s = 0, 1, \ldots$, случайных векторов называется случайной квазифейеровской последовательностью относительно множества G, если для произвольной точки $y\in G$

$$E(\|y - z^{s+1}\|^2/z^0, \ldots, z^s) \leqslant \|y - z^s\|^2 + W_s, \quad s = 0, 1, \ldots,$$

где случайные величины $W_s(\omega) \geqslant 0$ являются измеримыми относительно σ-подалгебры $\mathcal{B}_s$, индуцированной семейством $(z^0, \ldots, z^s)$ и такими, что

$$\sum_{s=0}^{\infty} EW_s < \infty.$$

Теорема 1'''. *Пусть имеют место предположения 0 и пусть η_k — случайная величина, измеримая относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами $(x^0, x^1, \ldots, x^k)$, такая, что для любого ω и некоторого числа c_ω*

$$E(\|\xi^k\|^2/x^0, \ldots, x^k) \leqslant \eta_k^2 \leqslant c_\omega \qquad (5.97)$$

при $\| x^s \| \leqslant w$, $s = 0, 1, \ldots, k$; *нормирующий множитель* γ_k *для некоторых чисел* $\bar{\gamma}$, $\underline{\gamma}$ *удовлетворяет условию*

$$0 < \underline{\gamma} \leqslant \gamma_k (\eta_k + \tau_k \| x^k \|) \leqslant \bar{\gamma} < \infty,$$

где $\tau_k = 1$, если $\| b^k \| > 0$ и $\tau_k = 0$, если $\| b^k \| = 0$; *величины* ρ_k, α_k, b^k *такие, что*

$$\rho_k \geqslant 0, \quad \alpha_k \geqslant 0, \quad \sum_{k=0}^{\infty} \boldsymbol{E} (\rho_k \| b^k \| + \rho_k^2) < \infty; \tag{5.98}$$

начальная точка x^0 *такая, что* $\boldsymbol{E} (\| x^0 \|)^2 < \infty$. *Тогда последовательность точек* x^k, $k = 0, 1, \ldots$, *порожденная алгоритмом 1, является случайной квазифейеровской относительно множества* X^*. *Если же, кроме того, с вероятностью 1*

$$\sum_{k=0}^{\infty} \rho_k \alpha_k = \infty,$$

то почти для каждого ω *последовательность* $\{x^k (\omega)\}_{k=0}^{\infty}$ *сходится к некоторому элементу* $x^* (\omega) \in X^*$.

Замечание 1. Условия теоремы 1‴ легко проверяются при решении конкретных задач. Заметим, что

$$\boldsymbol{E} (\| \xi^k \|^2 / \mathscr{B}_k) = \sum_{j=1}^{n} \boldsymbol{D} (\xi_j^k / \mathscr{B}_k) + \alpha_k^2 \| \hat{\nabla} f_0 (x^k) \|^2 + $$
$$+ 2\alpha_k (\hat{\nabla} f_0 (x^k), b^k) + \| b^k \|^2,$$

отсюда, например, следует, что если сумма дисперсий $\sum_{j=1}^{n} \boldsymbol{D} (\xi_j^k / \mathscr{B}_k)$ компонент вектора $\xi^k \triangleq (\xi_1^k, \ldots, \xi_n^k)$ ограничена в области X, а также ограничены α_k, $\| \hat{\nabla} f_0 (x^k) \|$, $\| b^k \|$, то $\eta_k = \text{const}$, т. е. (5.97) выполняется. Справедливость этого условия в реальных задачах обычно является следствием ограниченности области X.

Замечание 1′. Если область X ограничена, то утверждение теоремы 1‴ остается справедливым при $\gamma_k = \text{const}$.

Замечание 1″. Утверждение теоремы 1‴ остается справедливым также и при следующих (иногда полезных) изменениях условий: $\| b^k \|$ удовлетворяет условиям, аналогичным (5.97) т. е. $\| b^k \| \leqslant c_w$ при $\| x^s \| \leqslant w$, $s = 0, 1, \ldots, k$; нормирующий множитель γ_k удовлетворяет условиям

$$0 < \underline{\gamma} \leqslant \gamma_k \eta_k \leqslant r_k; \quad \underline{\gamma} \leqslant \gamma_k (1 + \| x^k \|) \| b^k \| \leqslant \delta_k,$$

где $\underline{\gamma}$ — некоторое число; δ_k, r_k — величины, измеримые относительно $\mathscr{B}_k$; вместо (5.98) справедливы условия

$$\rho_k \geqslant 0; \quad \alpha_k \geqslant 0; \quad \sum_{k=0}^{\infty} \boldsymbol{E} (\rho_k \delta_k + \rho_k^2 r_k^2) < \infty.$$

2. Стохастический метод сокращения невязок в детерминированных задачах

Задача 2. Найти $\arg\min_{x\in\mathscr{X}} f_0(x)$, где $\mathscr{X} = X \cap \{x \mid f_j(x) \leqslant 0, j = 1, \ldots, m\}$, $X \subset R^n$ — заданное множество, а $f_i : R^n \to R^1$, $i = 0, \ldots, m$ — заданные функции.

Предположения 2. (*i*) — функции $f_j(\cdot)$, $j = 0, 1, \ldots, m$ — непрерывные и выпуклые вниз в области X; (*ii*) — множество X выпуклое и замкнутое; (*iii*) — ограничения задачи 2 удовлетворяют условию Слейтера, поэтому функция Лагранжа

$$\varphi(x, u) = f_0(x) + \sum_{j=1}^{m} u_j f_j(x) \tag{5.99}$$

имеет седловую точку (x^*, u^*) в области $x \in X$, $u \geqslant 0$ ($u = (u_1, u_2, \ldots, u_m)$), причем множество $\{u^*\}$ компонент седловых точек $W = \{(x^*, u^*)\}$ ограничено.

Стохастический метод сокращения невязок является своеобразным стохастическим вариантом градиентного метода Эрроу — Гурвица. Однако, здесь не делается предположения о точном вычислении функций $f_j(x)$, $j = 0, 1, \ldots, m$. Стохастический метод сокращения невязок основан на применении функции Лагранжа (5.99), седловые точки которой вычисляются с помощью методов стохастических квазиградиентов.

Алгоритм 2

Начало. I. Выбрать начальную точку $(x^0, u^0) \in R^n \times R^m$, для которой $E(\|x^0\|^2 + \|u^0\|^2) < \infty$.

II. Выбрать последовательность шаговых множителей $\{\rho_k\}_{k=0}^{\infty}$ (измеримых относительно $\mathscr{B}_k$).

III. Выбрать последовательность нормирующих множителей $\{\gamma_k\}_{k=0}^{\infty}$ (измеримых относительно $\mathscr{B}_k$).

IV. Положить $k = 0$.

Основной цикл. V. Вычислить реализацию случайного вектора ξ^k, удовлетворяющего равенству

$$E(\xi^k/(x^0, u^0), \ldots, (x^k, u^k)) = \alpha_k \hat{\nabla}_x \varphi(x^k, u^k) + b^k,$$

где случайная величина $\alpha_k > 0$ и случайный вектор b^k измеримы относительно σ-подалгебры $\mathscr{B}_k$, индуцированной семейством величин $(x^0, u^0), \ldots, (x^k, u^k)$; $\hat{\nabla}_x \varphi(x^k, u^k)$ — вектор обобщенного градиента функции Лагранжа (5.99) по переменной x при фиксированном u в точке (x^k, u^k).

VI. Вычислить реализацию случайного вектора ζ^k, условное математическое ожидание которого

$$E(\zeta^k/(x^0, u^0), \ldots, (x^k, u^k)) = \alpha_k \nabla_u \varphi(x^k, u^k) + d^k,$$

где d^k— случайный вектор, измеримый относительно σ-подалгебры $\mathscr{B}_k$; $\nabla_u \varphi(x^k, u^k)$ — градиент функции Лагранжа по переменной

u при фиксированном x в точке (x^k, u^k), который равен вектору $(f_1(x^k), f_2(x^k), \ldots, f_m(x^k))$.

VII. Найти точку (x^{k+1}, u^{k+1}) по формулам

$$x^{k+1} = \pi_X(x^k - \rho_k\gamma_k\xi^k);$$
$$u^{k+1} = \pi_U(u^k + \rho_k\gamma_k\zeta^k),$$

где π_U — оператор проектирования на выпуклое и замкнутое множество U, содержащее компоненты u^* седловых точек (x^*, u^*) функции Лагранжа $\varphi(x, y)$ в области $x \in X$, $u \geqslant 0$.

VIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 2. *Пусть имеют место предположения 2 и, кроме того, $f_0(x)$ — строго выпуклая функция. Пусть η_k — случайная величина, измеримая относительно σ-подалгебры $\mathcal{B}_k$, индуцированной величинами $(x^0, u^0), \ldots, (x^k, u^k)$ такая, что*

$$E(\|\xi^k\|^2 + \|\zeta^k\|^2/(x^0, u^0), \ldots, (x^k, u^k)) \leqslant \eta_k^2 \leqslant c_w < \infty$$

при $\|x^s\| + \|u^s\| \leqslant w < \infty$, $s = 0, 1, \ldots, k$; нормирующий множитель γ_k для некоторых чисел $\underline{\gamma}$, $\overline{\gamma}$ удовлетворяет условию

$$0 < \underline{\gamma} \leqslant \gamma_k(\eta_k + \tau_k\|x^k\| + \theta_k\|u^k\|) \leqslant \overline{\gamma} < \infty,$$

где $\tau_k = 1$ при $\|b^k\| > 0$ и $\tau_k = 0$ при $\|b^k\| = 0$; $\theta_k = 1$ при $d^k\| > 0$ и $\theta_k = 0$ при $\|d^k\| = 0$; *величины ρ_k, α_k и векторы b^k, d^k такие, что*

$$\rho_k \geqslant 0, \quad \alpha_k \geqslant 0, \quad \sum_{k=0}^{\infty} E(\rho_k\|b^k\| + \rho_k\|d^k\| + \rho_k^2) < \infty. \qquad (5.100)$$

Тогда последовательность точек $\{(x^k, u^k)\}_{k=0}^{\infty}$, порождаемая алгоритмом 2, является случайной квазифейеровской последовательностью относительно множества седловых точек W. Если при этом с вероятностью 1

$$\sum_{k=0}^{\infty} \rho_k\alpha_k = \infty,$$

то с вероятностью 1 одна из предельных точек последовательности $\{x^k\}_{k=0}^{\infty}$ принадлежит X^*, т. е. почти наверное

$$\lim_{s\to\infty} (\min_{0\leqslant k\leqslant s} f_0(x^k)) = f_0(x^*), \quad x^* \in X^*.$$

Замечание 2. Утверждение теоремы 2 остается справедливым при условиях

$$\rho_k \geqslant 0, \quad \alpha_k \geqslant 0, \quad \sum_{k=0}^{\infty} E(\rho_k\delta_k + \rho_k^2 r_k^2) < \infty;$$

$$\|b^k\| + \|d^k\| \leqslant c_w \quad \text{при} \quad \|x^s\| + \|u^s\| \leqslant w, \quad s = 0, 1, \ldots, k;$$

нормирующий множитель γ_k удовлетворяет условиям

$$0 < \underline{\gamma} \leqslant \gamma_k\eta_k \leqslant r_k < \infty;$$

$$\underline{\gamma} \leqslant (1 + \|x^k\|)\|b^k\| + (1 + \|u^k\|)\|d^k\| \leqslant \delta_k,$$

где $\underline{\gamma}$ — некоторое число; величины r_k, δ_k измеримы относительно $\mathcal{B}_k$.

Замечание 2'. Если предположить, что функции $f_j(\cdot)$, $j = 0, 1, \ldots, m$, вычисляются точно и функция Лагранжа имеет по x вторые производные, ограниченные в области $x \in X$, $u \in U$, то на шаге V алгоритма 2 можно взять вектор

$$\xi^k = \sum_{s=1}^{s_k} \frac{\varphi(x^k + \Delta_k \beta^{k,s}, u^k) - \varphi(x^k, u^k)}{\Delta_k} \beta^{k,s},$$

где $\beta^{k,s}$, $s = 1, \ldots, s_k$ — серия независимых наблюдений вектора $\beta = (\beta_1, \ldots, \beta_n)$ с независимыми и равномерно распределенными на $[-1, 1]$ компонентами в k-й итерации, причем $s_k \geqslant 1$, $\Delta_k > 0$;

на шаге VI алгоритма 2 в качестве ζ^k можно взять вектор

$$\zeta^k = \frac{s_k}{3}(f_1(x^k), \ldots, f_m(x^k)).$$

Тогда

$$E(\xi^k/\mathcal{B}_k) = \frac{s_k}{3} \nabla_x \varphi(x^k, u^k) + v^k \Delta_k,$$

где $\| v^k \| \leqslant \text{const}$;

$$E(\zeta^k/\mathcal{B}_k) = \frac{s_k}{3} \nabla_u \varphi(x^k, u^k).$$

3. Метод сокращения невязок в задачах стохастического программирования

З а д а ч а 3. Найти $\arg\min_{x \in \mathcal{X}} f_0(x)$, где

$$\mathcal{X} \triangleq X \cap \{x \mid f_j(x) \leqslant 0, \quad j = 1, \ldots, m\};$$
$$f_j(x) \triangleq E F_j(x, \omega), \quad j = 0, 1, \ldots, m;$$

$X \subset R^n$ — заданное множество; $F_j : R^n \times \Omega \to R^1$ — заданные функции.

Для функций f_j, $j = 0, 1, \ldots, m$, и множества X выполняются предположения 2.

Алгоритм 3

Н а ч а л о. I. Выбрать произвольную начальную точку $(x^0, u^0) \in R^n \times R^m$.

II. Задать шаговый множитель ρ_0, нормирующий множитель γ_0 и величину смешения Δ_0, удовлетворяющие условиям теоремы 3.

III. Положить $k = 0$.

О с н о в н о й ц и к л. IV. Вычислить вектор

$$\xi^k = \sum_{i=1}^{n} \frac{\Phi(x^k + \Delta_k e^i, u^k, \omega^{k,i}) - \Phi(x^k, u^k, \omega^{k,0})}{\Delta_k} e^i,$$

где $\omega^{k,\nu}$, $\nu = 0, 1, \ldots, n$ — серия независимых по k наблюдений «состояния природы ω» (в частности, можно взять $\omega^{k,0} = \omega^{k,1} = \ldots$ $\ldots = \omega^{k,n} = \omega^k$);

функция $\Phi(x, u; \omega)$ определяется по правилу

$$\Phi(x, u, \omega) = F_0(x, \omega) + \sum_{j=1}^{m} u_j F_j(x, \omega),$$

здесь $u = (u_1, \ldots, u_m)$; e^i, $i = 1, \ldots, n$ — i-й орт.

Условное математическое ожидание вектора ξ^k равно

$$E(\xi^k/x^k, u^k) = \nabla_x \varphi(x^k, u^k) + v^k \Delta_k,$$

где функция

$$\varphi(x, u) \triangleq E\Phi(x, u, \omega);$$

v^k — некоторый вектор, для которого $\|v^k\| \leqslant \text{const}$, т. е. вектор ξ^k является стохастическим квазиградиентом функции $\varphi(x, u)$ в точке (x^k, u^k).

V. Вычислить вектор

$$x^{k+1} = \pi_X(x^k - \rho_k \gamma_k \xi^k).$$

VI. Вычислить вектор

$$u^{k+1} = \pi_U(u^k + \rho_k \gamma_k F(x^k, \omega^{k,0})),$$

где $F = (F_1, \ldots, F_m)$; множество U определено в пункте 2.

VII. Вычислить шаговый множитель ρ_{k+1}, нормирующий множитель γ_{k+1} и величину смещения по координатным осям Δ_{k+1}, удовлетворяющие условиям теоремы 3.

VIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 3. *Пусть выполнены предположения 2 и, кроме того, функции f_j, $j = 0, 1, \ldots, m$, имеют ограниченные в области X вторые производные и f_0 — строго выпуклая функция. Пусть:*

(i) — η_k — случайная величина, измеримая относительно σ-подалгебры, индуцированной величинами $(x^0, u^0), \ldots, (x^k, u^k)$, такая, что

$$E(\|\xi^k\|^2 + \|F(x^k, \omega^{k,0})\|^2/(x^0, u^0), \ldots, (x^k, u^k)) \leqslant \eta_k^2 \leqslant \beta(\chi) < \infty$$

при $\|x^s\| + \|u^s\| \leqslant \chi < \infty$, $s = 0, 1, \ldots, k$ (здесь $\beta(\chi)$ и χ — ограниченные константы);

(ii) — нормирующий множитель γ_k для некоторых чисел γ', γ'' удовлетворяет условию

$$0 < \gamma' \leqslant \gamma_k(\eta_k + \tau_k \|x^k\|) \leqslant \gamma'' < \infty,$$

где $\tau_k = 1$ при $\|v^k\| > 0$, $\tau_k = 0$ при $\|v^k\| = 0$;

(iii) — шаговый множитель ρ_k и величина смещения по координатным осям Δ_k детерминированные и такие, что

$$\rho_k \geqslant 0; \quad \sum_{k=0}^{\infty} \rho_k = \infty; \quad \sum_{k=0}^{\infty} (\rho_k |\Delta_k| + \rho_k^2) < \infty.$$

Тогда с вероятностью 1 одна из предельных точек последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, принадлежит X^*, т. е. почти наверное

$$\lim_{s\to\infty}(\min_{0\leqslant k\leqslant s} f_0(x^k)) = f_0(x^*), \quad x^*\in X^*.$$

4. Гибридный стохастический метод

Задача 4. Найти $\arg\min_{x\in X\cap A} f_0$ для заданных множеств $X\subset \subset R^n$ и $A = \{x|f(x)\leqslant 0\}$.

Предположения 4. (*i*) — функция $f_0(\cdot)$ — непрерывна и выпукла вниз; (*ii*) — множества X и A — выпуклые и замкнутые; (*iii*) — $A\cap X\neq\varnothing$.

Приводимый гибридный стохастический метод поиска экстремума функции $f_0(\cdot)$ сочетает в себе идеи стохастического релаксационного метода для решения систем неравенств и стохастического квазиградиентного метода для решения задач нелинейного программирования.

Алгоритм 4

Начало. I. Выбрать начальную точку $x^0\in R^n$, для которой $E\|x^0\|^2<\infty$.

II. Задать последовательности шаговых множителей

$$\{\delta_k\}_{k=0}^{\infty};\quad \{\rho_k\}_{k=0}^{\infty}.$$

III. Задать последовательности нормирующих множителей

$$\{\beta_k\}_{k=0}^{\infty};\quad \{\gamma_k\}_{k=0}^{\infty}.$$

IV. Положить $k=0$.

Основной цикл. V. Вычислить случайный вектор ξ^k, условное математическое ожидание которого

$$E(\xi^k/x^0, \ldots, x^k) = \alpha_k\hat{\nabla} f_0(x^k) + b^k, \tag{5.101}$$

где α_k— неотрицательная случайная величина и b^k — случайный вектор, измеримые относительно σ-подалгебры $\mathcal{B}_k$, индуцированной семейством случайных величин $(x^0, \ldots, x^k)$; $\hat{\nabla} f_0(x^k)$ — вектор обобщенного градиента функции f_0 в точке x^k.

VI. Вычислить случайный вектор ζ^k, условное математическое ожидание которого

$$E(\zeta^k/x^0, \ldots, x^k) = \lambda_k g(x^k) + d^k, \tag{5.102}$$

где λ_k — неотрицательная случайная величина и d^k — случайный вектор, измеримые относительно σ-подалгебры $\mathcal{B}_k$; $g(x^k)$ — вектор, для которого полупространство, отвечающее неравенству

$$(g(x), z-x) + f(x)\leqslant 0, \tag{5.103}$$

при $x=x^k$ содержит множество A, если $x^k\in A$.

VII. Найти вектор

$$x^{k+1} = \begin{cases} \pi_X (x^k - \delta_k \beta_k f(x^k) \zeta^k - \rho_k \gamma_k \xi^k), & f(x^k) > 0; \\ \pi_X (x^k - \rho_k \gamma_k \xi^k), & f(x^k) \leqslant 0. \end{cases}$$

VIII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 4. *Пусть имеют место предположения 4 и пусть η_k случайная величина, измеримая относительно σ-подалгебры $\mathcal{B}_k$, такая, что для любого числа $w < \infty$ найдется число c_w, для которого*

$$E(\|\xi^k\|^2 + \|\zeta^k\|^2 / x^0, \ldots, x^k) \leqslant \eta_k^2 \leqslant c_w$$

как только $\|x^s\| \leqslant w$, $s = 0, 1, \ldots, k$; для некоторых чисел, $\underline{\gamma}$, $\bar{\gamma}$ нормирующие множители β_k и γ_k удовлетворяют условиям

$$\beta_k (f(x^k)(\tau_1(k)\|x^k\| + 1) + \eta_k) = 1;$$

$$\underline{\gamma} \leqslant \gamma_k (\tau_2(k)\|x^k\| + \eta_k^2) \leqslant \bar{\gamma},$$

где $\tau_1(k) = 1$, если $\|d^k\| > 0$, $\tau_1(k) = 0$, если $\|d^k\| = 0$; $\tau_2(k) = 1$, если $\|b^k\| > 0$, $\tau_2(k) = 0$, если $\|b^k\| = 0$;
величины ρ_k, δ_k, α_k, λ_k и векторы b^k, d^k такие, что

$$\rho_k \geqslant 0, \quad 0 \leqslant \delta_k < 2\lambda_k - \varepsilon_k, \quad \varepsilon_k \geqslant 0, \quad \alpha_k \geqslant 0, \quad \lambda_k \geqslant 0;$$

$$\sum_{k=0}^{\infty} E(\rho_k \|b^k\| + \rho_k^2 + \rho_k \delta_k + \delta_k \|d^k\|) < \infty.$$

Тогда случайная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 4, является случайной квазифейеровской относительно множества X^*. Если же, кроме того, с вероятностью 1

$$\sum_{k=0}^{\infty} \delta_k \varepsilon_k = \infty; \quad \sum_{k=0}^{\infty} \rho_k \alpha_k = \infty,$$

то она сходится к некоторому элементу $x^* \in X^*$ почти наверное.

Замечание 4. Если, например,

$$f(x) = \max_i f_i(x) \triangleq f_{i(x)}(x),$$

где функции $f_i(x)$ — выпуклые вниз и непрерывно дифференцируемые; $i(x)$ — индекс, на котором достигается $\max\limits_i f_i(x)$ при заданном x, то вектор $g(x) = \nabla f_i(x)|_{i=i(x)}$ удовлетворяет неравенству (5.103).

Замечание 4'. Отметим, что на шаге VII алгоритма 4 используется стохастический релаксационный метод

$$x^{k+1} = \pi_X (x^k - \delta_k \beta_k f(x^k) \zeta^k)$$

для решения системы неравенств

$$f_i(x) \leqslant 0, \quad i = 1, \ldots, m;$$
$$x \in X,$$

где $f(x) = \max\limits_{1 \leqslant i \leqslant m} f_i(x)$.

Библиографические указания. При написании параграфа использовались работы [149, 153].

5.29. Комбинированный метод стохастических градиентов и штрафных функций

Задача 1. Найти $\arg\max_{x \in X} f_0(x)$ для заданной функции $f_0: R^n \to R^1$ и заданного множества

$$X = \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m, \quad x \in Y\}.$$

Предположения 1. (i) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — непрерывны вместе со своими производными; (ii) — функции $f_j(x)$, $j = 0, 1, \ldots, m$ — вогнуты; (iii) — Y — выпуклый компакт из R^n.

В приводимом здесь методе определяются функции

$$g_\tau(x, \alpha) = f_0(x) - \alpha \sum_{i=1}^{m} p_i |\min\{0, f_i(x)\}|^\tau$$

(здесь $\tau > 1$ — константа; α — коэффициенты штрафа, стремящиеся к бесконечности; p_i, $i = 1, \ldots, m$ — выбранные вероятности) и для отыскания их максимума применяется метод стохастического градиента.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^1 \in Y$.

II. Задать числа p_i, $i = 1, \ldots, m$ ($p_i > 0$, $i = 1, \ldots, m$; $\sum_{i=1}^{m} p_i = 1$), которые, соответственно, характеризуют требуемую относительную точность выполнения ограничений-неравенств в задаче 1 (обычно $p_i = 1/m$, $i = 1, \ldots, m$).

III. Выбрать параметр $\tau > 1$.

IV. Положить $k = 1$.

Основной цикл. V. Найти независимую реализацию i_k случайной величины $\tilde{i}$, которая принимает значения из множества $\mathcal{I} = \{1, \ldots, m\}$, соответственно, с вероятностями $p_1, \ldots, p_m$.

VI. Найти значения шагового множителя ρ_k и коэффициента штрафа α_k, удовлетворяющие условиям теоремы 1.

VII. Вычислить стохастический градиент ξ^k функции $g_\tau(x, \alpha_k)$ в точке x^k

$$\xi^k = \nabla f_0(x^k) + \alpha_k \tau |\min\{0, f_{i_k}(x^k)\}|^{\tau-1} \nabla f_{i_k}(x^k).$$

VIII. Вычислить следующее приближение

$$x^{k+1} = \pi_Y(x^k + \rho_k \xi^k).$$

IX. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Если выполняются предположения 1, множество X — непусто и числовые последовательности $\{\rho_k\}_{k=1}^{\infty}$, $\{\alpha_k\}_{k=1}^{\infty}$ удовлетво-*

рают условиям

$$\rho_k > 0, \quad \lim_{k\to\infty} \rho_k = 0, \quad \rho_{k+1} \leqslant \rho_k, \quad \sum_{k=1}^{\infty} \rho_k = \infty;$$

$$\alpha_k > 0, \quad \lim_{k\to\infty} \alpha_k = \infty, \quad \alpha_{k+1} \geqslant \alpha_k, \quad \sum_{k=1}^{\infty} (\rho_k \alpha_k)^2 < \infty,$$

то для любого начального приближения x^1 *последовательность* $\{x^k\}_{k=1}^{\infty}$, *порождаемая алгоритмом 1, с вероятностью 1 сходится к* $\operatorname{Arg}\max_{x\in X} f_0(x)$ — *множеству решений задачи 1.*

Замечание 1. Алгоритм 1 в [372] рекомендуется применять для решения задач математического программирования с большим числом ограничений (в особенности для тех задач, ограничения которых могут формироваться по мере надобности в ЭВМ) и для решения задач с блочной структурой.

Библиографические указания. При написании параграфа использовалась работа [372].

5.30. Методы усреднения направлений спуска

1. Детерминированная задача

Задача 1. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0 : \boldsymbol{R}^n \to \boldsymbol{R}^1$ и множества X, заданного соотношением

$$X = X_1 \cap X_2,$$

где $X_1 = \{x \mid f_j(x) \leqslant 0, \ j = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}$; X_2 — выпуклое, замкнутое, ограниченное множество.

Предположения 1. (*i*) — функции f_j, $j = 0, 1, \ldots, m$ — выпуклы; (*ii*) — множество X_2 имеет непустое пересечение с множеством внутренних точек множества X_1.

Ниже приводятся два алгоритма усреднения направлений спуска, первый из которых требует вычисления субградиентов функций f_j, $j = 1, \ldots, m$, а второй — лишь вычисления значений функций f_j, $j = 1, \ldots, m$.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение

$$x^0 \in \boldsymbol{R}^n.$$

II. Выбрать произвольное натуральное число l.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям теоремы 1.

V. Если выполняется неравенство

$$\max_{1\leqslant j\leqslant m} f_j(x^k) \leqslant 0,$$

то положить $\nu_k = 0$ и перейти к шагу VII, иначе перейти к шагу VI.

VI. Вычислить индекс $i \in [1 : m]$, удовлетворяющий условию

$$f_i(x^k) = \max_{1 \leqslant j \leqslant m} f_j(x^k),$$

положить $\nu_k = i$ и перейти к шагу VII.

VII. Вычислить субградиент $\hat{\nabla} f_{\nu_k}(x^k)$ функции f_{ν_k} в точке x^k.

VIII. Если $k \geqslant l$, то вычислить вектор

$$h^k = \sum_{i=k-l}^{k} \hat{\nabla} f_{\nu_i}(x^i)$$

и перейти к шагу IX; иначе вычислить вектор h^k по формуле

$$h^k = \sum_{i=0}^{k} \hat{\nabla} f_{\nu_i}(x^i)$$

и перейти к шагу IX.

IX. Вычислить следующее приближение

$$x^{k+1} = \pi_{X_2}(x^k - \rho_k h^k),$$

где $\pi_{X_2}(x)$ — оператор проектирования на множество X_2 точки $x \in R^n$.

X. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Если выполнены предположения 1 и условия*

$$\rho_k \geqslant 0, \quad k = 0, 1, \ldots; \quad \sum \rho_k = \infty; \quad \rho_k \to 0 \quad при \quad k \to \infty,$$

то при любом начальном приближении $x^0 \in R^n$ найдется подпоследовательность $\{x^{k_i}\}_{i=0}^{\infty}$ последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, такая, что

$$\lim_{i \to \infty} f_0(x^{k_i}) = \min_{x \in X} f_0(x).$$

Если, кроме того, f_0 — строго выпуклая функция, то

$$\lim_{k \to \infty} x^k = x^*,$$

где

$$f_0(x^*) = \min_{x \in X} f_0(x), \quad x^* \in X.$$

Алгоритм 1'

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$.

II. Выбрать произвольное натуральное число l.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить шаговый множитель ρ_k и параметры δ_k, α_k, удовлетворяющие условиям теоремы 1'.

V. Если выполняется неравенство

$$\max_{1 \leqslant j \leqslant m} f_j(x^k) \leqslant 0,$$

то положить $\nu_k = 0$ и перейти к шагу VII; иначе перейти к шагу VI.

VI. Вычислить индекс $i \in [1:m]$, удовлетворяющий условию

$$f_i(x^k) = \max_{1 \leqslant j \leqslant m} f_j(x^k),$$

положить $\nu_k = i$ и перейти к шагу VII.

VII. Вычислить реализацию $\tilde{x}^k$ случайной точки, равномерно распределенной в n-мерном кубе с центром в точке x^k и стороной α_k.

VIII. Вычислить вектор

$$\theta^k = \sum_{i=1}^{n} \frac{f_{\nu_k}(\tilde{x}^k + \delta_k e^i) - f_{\nu_k}(\tilde{x}^k)}{\delta_k} e^i,$$

где e^i, $i = 1, \ldots, n$ — i-й орт.

IX. Если $k \geqslant l$, то вычислить вектор

$$h^k = \sum_{i=k-l}^{k} \theta^i$$

и перейти к шагу X; иначе вычислить вектор h^k по формуле

$$h^k = \sum_{i=0}^{k} \theta^i$$

и перейти к шагу X.

X. Вычислить следующее приближение

$$x^{k+1} = \pi_{X_2}(x^k - \rho_k h^k).$$

XI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1'. *Если выполнены предположения 1 и условия*

$$\rho_k \geqslant 0, \quad k = 0, 1, \ldots, \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty;$$

$$\alpha_k \geqslant 0, \quad k = 0, 1, \ldots;$$

$$\delta_k/\alpha_k \to 0 \quad \text{при} \quad k \to \infty, \quad \alpha_k \to 0 \quad \text{при} \quad k \to \infty,$$

то с вероятностью 1 существует подпоследовательность $\{x^{k_i}\}_{i=0}^{\infty}$ последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, такая, что

$$\lim_{i \to \infty} f_0(x^{k_i}) = \min_{x \in X} f_0(x).$$

Если, кроме того, f_0 — строго выпуклая функция, то с вероятностью 1

$$\lim_{k \to \infty} x^k = x^*,$$

где

$$f_0(x^*) = \min_{x \in X} f_0(x), \quad x^* \in X.$$

2. Стохастическая задача

Задача 2. Найти

$$\arg\min_{x \in X} EF_0(x, \omega) \quad \left(EF_0(x, \omega) \triangleq \int_\Omega F_0(x, \omega)\, \mu(d\omega)\right)$$

для заданной функции $F_0 : X \times \Omega \to R^1$ и заданного множества $X \subset R^n$.

Предположения 2. (*i*) — функция f_0 — непрерывно дифференцируема и выпукла вниз в области X; (*ii*) — X — выпуклое и замкнутое множество.

В приводимом непоисковом алгоритме адаптивного управления на k-й ($k \geqslant p_0$) итерации делается только одно измерение (вычисление) функции F_0. Для вычисления направления движения ξ^k к следующему приближению x^{k+1} используется известная информация из p_k ($p_k \geqslant 1$) предыдущих итераций

$$\xi^k = \sum_{s=k-p_k+1}^{k} \frac{1}{\Delta_s} (F_0(x^s + \Delta_s \theta^s, \omega^s) - F_0(x^{s-1} + \Delta_{s-1}\theta^{s-1}, \omega^{s-1}))\, \theta^s,$$

где $\Delta_k > 0$; $\{\theta^s\}_{s=k-p_k}^{k}$ — серия независимых наблюдений случайного вектора θ с независимыми и равномерно распределенными на $[-1, 1]$ компонентами.

Алгоритм 2

Начало. I. Выбрать начальное значение параметра p_0, удовлетворяющее условиям теоремы 2.

II. Найти: начальные точки x^k, $k = 0, 1, \ldots, p_0$, принадлежащие множеству X; векторы θ^k, $k = 0, 1, \ldots, p_0$ — серию независимых наблюдений случайного вектора θ с независимыми и равномерно распределенными на отрезке $[-1, 1]$ компонентами; реализации ω^k, $k = 0, 1, \ldots, p_0$ случайного события ω.

III. Задать смещения Δ_k, $k = 0, 1, \ldots, p_0$, удовлетворяющие условиям теоремы 2.

IV. Положить $k = p_0$.

Основной цикл. V. Вычислить параметр p_k и вектор

$$\xi^k = \sum_{i=k-p_k+1}^{k} \frac{1}{\Delta_i} (F_0(x^i + \Delta_i \theta^i, \omega^i) - F_0(x^{i-1} + \Delta_{i-1}\theta^{i-1}, \omega^{i-1}))\, \theta^i.$$

VI. Вычислить шаговый множитель ρ_k и нормирующий множитель γ_k, удовлетворяющие условиям теоремы 2.

VII. Вычислить следующее приближение

$$x^{k+1} = \pi_X (x^k - \rho_k \gamma_k \xi^k).$$

VIII. Найти независимое наблюдение θ^{k+1} случайного вектора θ с независимыми и равномерно распределенными на отрезке $[-1, 1]$ компонентами.

IX. Найти реализацию ω^{k+1} случайного события ω.

X. Найти смещение Δ_{k+1}, удовлетворяющее условиям теоремы 2.

XI. Положить $k = k + 1$ и перейти к шагу V.

Теорема 2. *Пусть выполнены предположения 2 и имеют место условия:* (*i*) — *градиент функции* f_0 *удовлетворяет глобальному условию Липшица, т. е. при* $x, y \in R^n$

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \alpha_1 \|x - y\|, \quad \alpha_1 < \infty;$$

(*ii*) — *известна величина* β_k *такая, что*

$$E\left(\sum_{i=k-p_k+1}^{k} \|(F_0(x^i + \Delta_i\theta^i, \omega^i) - \right.$$

$$\left. - F_0(x^{i-1} + \Delta_{i-1}\theta^{i-1}, \omega^{i-1}))\,\theta^i\|^2 / x^{k-p_k+1}, \ldots, x^k\right) \leqslant \beta_k^2 \leqslant \delta(\tau) < \infty$$

при $\|x^s\| \leqslant \tau < \infty$, $s = k - p_k, \ldots, k$; (*iii*) — *нормирующий множитель* γ_k *удовлетворяет условию*

$$0 < \gamma' \leqslant \gamma_k(\|x^k\| + \beta_k) \leqslant \gamma'' < \infty;$$

(*iv*) — *величина* Δ_k, p_k и ρ_k *такие, что*

$$1 \leqslant p_k \leqslant \alpha_2 < \infty, \quad k - p_k \geqslant 0, \quad \rho_k \geqslant 0, \quad \Delta_k > 0;$$

$$\rho_k \to 0, \quad \Delta_k \to 0 \quad \text{при} \quad k \to \infty;$$

$$\sum_{k=p_0}^{\infty} \rho_k \Delta_{k-p_k+1} < \infty, \quad \sum_{k=0}^{\infty} (\rho_k/\Delta_k)^2 < \infty.$$

Тогда последовательность случайных точек $\{x^k(\omega)\}_{k=0}^{\infty}$, *порожденная алгоритмом 2, является случайной квазифейеровской относительно множества решений* X^* *задачи 2.*

Если, кроме того, выполняется условие (*v*) — $\sum_{k=0}^{\infty} \rho_k = \infty$, *то почти для каждого* ω *последовательность* $\{x^k(\omega)\}_{k=0}^{\infty}$ *сходится к решению задачи 2.*

Библиографические указания. Пункт 1 написан на основании работ [96, 97, 308, 309, 379, 380], пункт 2 основан на результатах работы [53].

5.31. Прямой метод решения задач стохастического программирования

З а д а ч а 1. Найти $\arg\min_{x \in X} E_\omega f_0(x, \omega)$ для заданной функции $f_0 : R^n \times \Omega \to R^1$ и заданного множества

$$X = \{x \mid E_\omega f_i(x, \omega) \leqslant 0, \quad i = 1, \ldots, m, \quad x \in X^1\},$$

где X^1 — некоторое множество пространства R^n.

Предположения 1. (*i*) — X^1 — выпуклое, замкнутое и ограниченное множество; (*ii*) — $E_\omega f_i(x, \omega)$, $i = 0, 1, \ldots, m$ — непрерывные, выпуклые вниз функции.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in R^n$; произвольные числа z_i^0, $i = 1, \ldots, m$; достаточно большую константу $\delta > 0$.

II. Положить $k = 0$.

Основной цикл. III. Найти шаговый множитель ρ_k и множитель α_k, удовлетворяющие условиям теоремы 1.

IV. Если выполняется условие

$$\max_{1 \leqslant i \leqslant m} z_i^k \leqslant 0,$$

то положить $j = 0$ и перейти к шагу VI; иначе перейти к шагу V.

V. Найти индекс $j \in [1 : m]$, удовлетворяющий условию

$$z_j^k = \max_{1 \leqslant i \leqslant m} z_i^k.$$

VI. Вычислить реализацию ξ^k случайного вектора $\bar{\xi}^k$, для которого

$$E_\omega (\bar{\xi}^k / x^0, z^0, \ldots, x^k, z^k) = \hat{\nabla} \varphi_j (x^k),$$

где $\hat{\nabla} \varphi_j (x^k)$ — обобщенный градиент функции $\varphi_j (x) \triangleq E_\omega f_j (x, \omega)$ в точке $x = x^k$.

VII. Вычислить следующее приближение

$$x^{k+1} = \pi_{X^1} (x^k - \rho_k \xi^k).$$

VIII. Найти случайные величины θ_i^k, $i = 1, \ldots, m$, для которых

$$E (\theta_i^k / x^0, z^0, \ldots, x^k, z^k) = E_\omega f_i (x^k, \omega), \quad i = 1, \ldots, m.$$

IX. Положить $i = 1$.

X. Если выполняется неравенство

$$| z_i^k + \alpha_k (\theta_i^k - z_i^k) | \leqslant \delta,$$

то положить

$$z_i^{k+1} = z_i^k + \alpha_k (\theta_i^k - z_i^k)$$

и перейти к шагу XI; иначе положить

$$z_i^{k+1} = (\delta z_i^k + \delta \alpha_k (\theta_i^k - z_i^k)) / | z_i^k + \alpha_k (\theta_i^k - z_i^k) |$$

и перейти к шагу XI.

XI. Если $i < m$, то положить $i = i + 1$ и перейти к шагу X; иначе перейти к шагу XII.

XII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Если выполнены предположения 1 и, кроме того:* (***iii***) *— область Y, определяемая ограничениями*

$$E_\omega f_i (x, \omega) \leqslant 0, \quad i = 1, \ldots, m,$$

удовлетворяет условию Слейтера; (***iv***) *—*

$$\sum_{k=0}^{\infty} \rho_k = \infty, \quad \sum_{k=0}^{\infty} \rho_k^2 < \infty, \quad \rho_k / \alpha_k \to 0 \quad \textit{при} \quad k \to \infty;$$

(v) —

$$E|\theta_i^k| \leqslant \delta < \infty, \quad i = 1, \ldots, m, \quad \text{при} \quad k = 0, 1, \ldots;$$

(vi) —

$$E\|\xi^k\|^2 \leqslant \delta < \infty \quad \text{при} \quad k = 0, 1, \ldots,$$

то с вероятностью 1 одна из предельных точек последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, принадлежит множеству решений задачи 1. Если* $Ef_0(x, \omega)$ *— строго выпуклая вниз функция, то с вероятностью 1*

$$\lim_{k\to\infty} x^k = x^*,$$

где x^* *— решение задачи 1.*

Замечание 1. Если функции $f_i(x, \omega)$, $i = 0, 1, \ldots, m$, при каждом ω выпуклые вниз, то на шаге VI алгоритма 1 можно взять

$$\xi^k = \hat{\nabla} f_j(x^k, \omega^k),$$

а на шаге VIII —

$$\theta_i^k = f_i(x^k, \omega^k), \quad i = 1, \ldots, m,$$

где $\hat{\nabla} f_j(x^k, \omega^k)$ — обобщенный градиент функции $f_j(x, \omega)$ по x; ω^k — независимые наблюдения состояния природы.

Библиографические указания. При написании параграфа использовались работы [92, 97].

5.32. Метод случайного поиска в выпуклых задачах минимизации

Задача 1. Найти $\arg\min_{x\in X} f_0(x)$ для заданной функции $f_0: R^n \to R^1$ и заданного множества $X \subset R^n$.

Предположения 1. (*i*) — функция $f_0(x)$ — выпукла и непрерывно дифференцируема на множестве X; (*ii*) — градиент функции $f_0(x)$ удовлетворяет на X условию Липшица с константой $0 < \gamma < \infty$, т. е.

$$\|\nabla f_0(x) - \nabla f_0(y)\| \leqslant \gamma \|x - y\|, \quad \forall x, y \in X;$$

(*iii*) — множество X — выпукло и замкнуто.

В методе случайного поиска на k-й итерации по известному приближению $x^k \in X$ вычисляется следующее приближение x^{k+1}, как точка в некотором смысле близкая к точке минимума функции $f_0(\cdot)$ на отрезке прямой $x^k - \rho h^k$ ($\rho \in (-\infty, \infty)$), принадлежащем множеству X, где h^k — независимая реализация единичного случайного вектора, равномерно распределенного на единичной сфере с центром в начале координат. Для вычисления шагового множителя ρ_k необходимо применять конечное число итераций какого-нибудь алгоритма одномерной оптимизации.

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $x^0 \in X$.

II. Положить $k = 0$.

Основной цикл. III. Если $\nabla f_0(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу IV.

IV. Найти независимую реализацию h^k единичного случайного вектора ξ, равномерно распределенного на единичной n-мерной сфере с центром в начале координат.

V. Если множество

$$L_k = \{\rho \mid x^k - \rho h^k \in X, \quad \rho \in (-\infty, \infty)\},$$

содержит хотя бы одну точку $\rho \neq 0$, то перейти к шагу VI; иначе положить $x^{k+1} = x^k$ и перейти к шагу VIII.

VI. Вычислить шаговый множитель $\rho_k \in L_k$, удовлетворяющий условию

$$f_0(x^k - \rho_k h^k) \leqslant (1 - \lambda_k) f_0(x^k) + \lambda_k \omega_k,$$

где λ_k — произвольная точка фиксированного отрезка $[\lambda, 1]$; λ — произвольная константа из полуинтервала $(0, 1]$:

$$\omega_k = \min_{\rho \in L_k} f_0(x^k - \rho h^k).$$

VII. Вычислить следующее приближение

$$x^{k+1} = x^k - \rho_k h^k.$$

VIII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются предположения 1 и* (**iv**) — *множество X регулярно, т. е. в $\boldsymbol{R}^n$ существует сфера радиуса χ, целиком принадлежащая множеству X;* (**v**) — *для любого элемента $x^0 \in X$ множество*

$$X_0 = \{x \mid f_0(x) \leqslant f_0(x^0), \quad x \in X\}$$

ограничено. Тогда бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 1, такова, что

$$\boldsymbol{P}\{f_0(x^k) \to f_0(x^*), \quad k \to \infty\} = 1.$$

Библиографические указания. Параграф написан на основании работы [193].

5.33. Методы решения задач оптимизации с бесконечным числом ограничений

Задача 0. Найти $\arg\min_{x \in X} f_0(x)$ для заданной функции $f_0: \boldsymbol{R}^n \to \boldsymbol{R}^1$ и заданного компактного множества $X \subset \boldsymbol{R}^n$ при дополнительных ограничениях

$$f_1(x, y) \leqslant 0 \text{ для всех } y \in Y,$$

где $f_1: \boldsymbol{R}^n \times \boldsymbol{R}^m \to \boldsymbol{R}^1$, $Y \subset \boldsymbol{R}^m$.

Многие задачи оптимизации могут быть представлены в форме задачи 0. Например, если в минимаксной задаче вида

$$v^* = \min_{x \in X} \max_{y \in Y} \varphi(x, y)$$

положить $v_1 < v^*$, $v_2 > v^*$ и $X_0 = [v_1, v_2]$, то ее можно представить в форме задачи 0 как $\min_{(x_0, x) \in X_0 \times X} x_0$ при ограничении $\varphi(x, y) - x_0 \leqslant 0$ для всех $y \in Y$; другие примеры можно найти в [435].

Предположения 0. (*i*) — функции f_0, f_1 являются непрерывными на множествах X и $X \times Y$, соответственно; (*ii*) — X и Y — непустые компактные множества.

На k-й итерации находят приближение x^k как решение аппроксимирующей задачи:

$$\text{найти } \arg\min_{x \in G_k} f_0(x),$$

где

$$G_k = \{x \mid f_1(x, y) \leqslant 0 \text{ для всех } y \in Y_k,\ x \in X\}.$$

Конечные множества Y_k аппроксимируют множество Y и их находят с помощью решения задачи максимизации по $y \in Y$ функции $f_1(x^k, y)$.

1. Общий алгоритм

Алгоритм 1

Начало. I. Выбрать произвольное конечное подмножество $Y_0 \subset Y$.

II. Положить $k = 0$.

Основной цикл. III. Найти допустимое множество G_k k-й аппроксимирующей задачи по правилу

$$G_k = \{x \mid f_1(x, y) \leqslant 0 \text{ для всех } y \in Y_k,\ x \in X\}.$$

IV. Найти решение $x^k \in G_k$ следующей k-й аппроксимирующей задачи:

$$\text{найти } \arg\min_{x \in G_k} f_0(x).$$

(Отметим, что выполняются включения

$$G \subset \cdots \subset G_{k+1} \subset G_k \subset \cdots \subset G_0, \tag{5.104}$$

где $G \triangleq \{x \mid f_1(x, y) \leqslant 0$ для всех $y \in Y,\ x \in X\}$, и справедливы неравенства

$$f_0(x^0) \leqslant f_0(x^1) \leqslant \cdots \leqslant f_0(x^k) \leqslant \cdots \leqslant f_0(x^*), \tag{5.105}$$

где x^* — решение задачи 0).

V. Найти решение $y^{k+1} \in Y$ следующей k-й вспомогательной задачи:

$$\text{найти } \arg\max_{y \in Y} f_1(x^k, y).$$

VI. Если выполняется неравенство

$$f_1(x^k, y^{k+1}) \leqslant 0,$$

то положить $x^* = x^k$ и прекратить вычисления (решение задачи 0 находится в этом случае за конечное число итераций); иначе положить $Y_{k+1} = Y_k \cup \{y^{k+1}\}$ и перейти к шагу VII.

VII. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняется предположение 0. Если x^* есть предельная точка бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, то x^* является решением задачи 0.*

Теорема 1'. *Пусть выполняются предположения 0 и (iii) — функция $f_1(x, y)$ — выпукла по y при всех $x \in X$; (iv) — множество Y является многоугольником. Тогда алгоритм 1 сходится к решению x^* задачи 0 за конечное число итераций.*

2. Ослабленный алгоритм

В ослабленном алгоритме на каждой итерации отбрасываются все несвязные ограничения. Сходимость ослабленного алгоритма имеет место при некоторых дополнительных предположениях.

Алгоритм 2

I—V (шаги I—V такие, как и в алгоритме 1).

Если выполняются только предположения 0, то соотношения (5.104) и (5.105) для алгоритма 2 в общем не имеют места.

VI. Если выполняется неравенство

$$f_1(x^k, y^{k+1}) \leqslant 0,$$

то положить $x^* = x^k$ и прекратить вычисления (в этом случае решение x^* задачи 0 находят за конечное число итераций); иначе перейти к шагу VII.

VII. Положить

$$Z_k = \{y \mid f_1(x^k, y) < 0, \ y \in Y_k\}.$$

VIII. Положить

$$Y_{k+1} = Y_k \cup \{y^{k+1}\} \diagdown Z_k.$$

IX. Положить $k = k + 1$ и перейти к шагу III.

Теорема 2. *Пусть выполняются предположения 0. Тогда:*

1) *если бесконечная последовательность $\{x^k\}_{k=0}^{\infty}$, порожденная алгоритмом 2, сходится к некоторой точке x^*, то x^* является решением задачи 0;*

2) *если функции $f_0(x)$ и $f_1(x, y)$ выпуклы по x (при каждом $y \in Y$), а X — выпуклое множество, то последовательность $\{f_0(x^k)\}_{k=0}^{\infty}$ монотонно не убывает;*

3) *если функция $f_0(x)$ строго выпукла, каждая из функций $f_1(x, y)$ выпукла по x для $y \in Y$, X — выпуклое множество, то последовательность $\{x^k\}_{k=0}^{\infty}$ сходится к решению x^* задачи 0.*

Библиографические указания. При написании параграфа использовалась работа [435].

5.34. Методы решения задач квадратичного программирования

В этом параграфе приводятся некоторые методы решения задачи квадратичного программирования, которая является вспомогательной задачей для многих алгоритмов. Задачей квадратичного программирования называют задачу минимизации квадратичной функции при линейных ограничениях.

1. Метод сопряженных градиентов для минимизации квадратичной функции на подпространстве

Задача 1. Найти $\arg\min\limits_{x \in X} \left[\frac{1}{2} (Cx, x) + (d, x) \right]$,

где

$$X = \{x \mid (a^i, x) - b_i = 0, \quad i \in \mathcal{I}\};$$

C — симметричная положительно определенная матрица размера $n \times n$; $d \in \boldsymbol{R}^n$, $a^i \in \boldsymbol{R}^n$, $i \in \mathcal{I}$, $\mathcal{I}$ — конечное множество индексов; $b_i \in \boldsymbol{R}^1$, $i \in \mathcal{I}$.

Предположения 1. (*i*) — C — симметричная положительно определенная матрица размера $n \times n$, т. е. $(Cx, x) > 0$ для всех $x \in \boldsymbol{R}^n$; (*ii*) — векторы a^i, $i \in \mathcal{I}$,— линейно-независимы.

Алгоритм 1

Начало. I. Найти начальное приближение $x^0 \in X$.

II. Вычислить $n \times n$-матрицу H по формуле

$$H = A^T (AA^T)^{-1} A,$$

где A — матрица, строками которой являются n-мерные вектор-строки $(a^i)^T$, $i \in \mathcal{I}$.

III. Вычислить градиент минимизируемой функции

$$f_0(x) \triangleq \frac{1}{2} (Cx, x) + (d, x)$$

в точке $x = x^0$ по формуле

$$\nabla f_0(x^0) = Cx^0 + d.$$

IV. Вычислить n-мерный вектор

$$h^1 = -(I - H) \nabla f_0(x^0),$$

где I — единичная $n \times n$-матрица.

Если $h^1 = 0$, то прекратить вычисления, иначе перейти к шагу V.

V. Вычислить шаговый множитель

$$\rho_1 = -(\nabla f_0(x^0), h^1)/(h^1, Ch^1).$$

VI. Положить $x^1 = x^0 + \rho_1 h^1$.

VII. Положить $k = 1$, $g^0 = h^1$.

Основной цикл. VIII. Вычислить n-мерный вектор

$$g^k = (I - H) \nabla f_0(x^k),$$

где $\nabla f_0(x^k) = Cx^k + d$.

Если $g^k = 0$, то прекратить вычисления; иначе перейти к шагу IX.

IX. Вычислить n-мерный вектор

$$h^{k+1} = -g^k + \frac{\|g^k\|^2}{\|g^{k-1}\|^2} h^k.$$

X. Вычислить шаговый множитель

$$\rho_{k+1} = -(\nabla f_0(x^k),\ h^{k+1})/(h^{k+1},\ Ch^{k+1}).$$

XI. Положить $x^{k+1} = x^k + \rho_{k+1} h^{k+1}$.

XII. Положить $k = k + 1$ и перейти к шагу VIII.

Теорема 1. *Если выполнены предположения 1, то алгоритм 1 решает задачу 1 за конечное число итераций, не превышающее n. При этом, если минимизируемая функция $f_0(x) \triangleq \frac{1}{2}(Cx, x) + (d, x)$ имеет на множестве X конечный минимум, то последовательность $\{x^k\}$, порожденная алгоритмом 1, сходится к точке минимума x^* задачи 1 за конечное число итераций $k \leqslant n$. Если функция $f_0(x)$ неограничена (снизу) на множестве X, то при некотором $k \leqslant n$ будет выполняться равенство*

$$(h^{k+1},\ Ch^{k+1}) = 0,$$

что влечет $\rho_{k+1} = -\infty$.

2. Метод сопряженных градиентов для общей задачи квадратичного программирования

Задача 2. Найти $\arg\min\limits_{x \in X} \left[\frac{1}{2}(Cx,\ x) + (d,\ x)\right]$,

где

$$X \triangleq \{x \mid (a^i,\ x) - b_i \leqslant 0,\ i \in \mathcal{I}^-;\ \ (a^i,\ x) - b_i = 0,\ i \in \mathcal{I}^0\};$$

C — симметричная положительно определенная матрица размера $n \times n$; $d \in \boldsymbol{R}^n$; $\mathcal{I}^-$ и $\mathcal{I}^0$ — конечные множества индексов; $a^i \in \boldsymbol{R}^n$, $i \in \mathcal{I}^0 \cup \mathcal{I}^-$; $b_i \in \boldsymbol{R}^1$, $i \in \mathcal{I}^0 \cup \mathcal{I}^-$.

Предположения 2. (*i*) — C — симметричная положительно определенная матрица; (*ii*) — выполнено следующее условие невырожденности: при любом $x \in \boldsymbol{R}^n$ векторы a^i, $i \in \mathcal{I}(x)$ — линейно-независимы, где

$$\mathcal{I}(x) \triangleq \{i \mid (a^i,\ x) - b_i = 0,\quad i \in \mathcal{I}^- \cup \mathcal{I}^0\}. \qquad \textbf{(5.106)}$$

Алгоритм 2

Начало. I. Найти начальное приближение $x^0 \in X$.

II. Найти множество индексов $\mathcal{I}(x^0)$ и положить $\mathcal{I}_0 = \mathcal{I}(x^0)$ (здесь множество $\mathcal{I}(x)$ определяется по (5.106)).

III. Вычислить $n \times n$-матрицу

$$H_{\mathcal{I}_0} = A^T_{\mathcal{I}_0}(A_{\mathcal{I}_0} A^T_{\mathcal{I}_0})^{-1} A_{\mathcal{I}_0},$$

где $A_{\mathcal{I}_0}$ — матрица, строками которой являются n-мерные вектор-строки $(a^i)^T$, $i \in \mathcal{I}_0$.

IV. Вычислить вектор

$$u^0 = -(A_{\mathcal{J}_0} A_{\mathcal{J}_0}^T)^{-1} A_{\mathcal{J}_0} \nabla f_0(x^0),$$

где $\nabla f_0(x) = Cx + d$.

V. Вычислить n-мерный вектор

$$g^0 = (I - H_{\mathcal{J}_0}) \nabla f_0(x^0) = \nabla f_0(x^0) + A_{\mathcal{J}_0}^T u^0. \tag{5.107}$$

Если $g^0 = 0$, то перейти к шагу VI; иначе перейти к шагу XXII. (В случае, когда $g^0 = 0$, точка x^0 является точкой минимума функции $f_0(x) \triangleq \frac{1}{2}(Cx, x) + (d, x)$ на грани, определяемой системой уравнений $(a^i, x) - b_i = 0$, $i \in \mathcal{J}_0$).

VI. Определить компоненты вектора u^0, которые в равенствах (5.107) умножаются на векторы a^i, $i \in \mathcal{J}^- \cap \mathcal{J}_0$. Пусть они будут $u_{j1}^0, u_{j2}^0, \ldots, u_{jm_0}^0$.

VII. Если $u_{j1}^0 \geqslant 0, \ldots, u_{jm}^0 \geqslant 0$, то положить $x^* = x^0$ и прекратить вычисления (в этом случае x^* — решение задачи 2); иначе перейти к шагу VIII.

VIII. Найти индекс $j_{m'} \in \{j_1, \ldots, j_m\}$, для которого выполняется неравенство $u_{j_{m'}}^0 < 0$.

IX. Найти индекс $i' \in \mathcal{J}_0 \cap \mathcal{J}^-$ такой, что вектор $a^{i'}$ в (5.107) умножается на компоненту $u_{j_{m'}}^0$.

X. Образовать новое индексное множество $\mathcal{J}_0'$, которое получается из множества $\mathcal{J}_0$ отбрасыванием индекса i'.

XI. Образовать матрицу $A_{\mathcal{J}_0'}$, строками которой являются n-мерные вектор-строки $(a^i)^T$, $i \in \mathcal{J}_0'$.

XII. Вычислить $n \times n$-матрицу

$$H_{\mathcal{J}_0'} = A_{\mathcal{J}_0'}^T (A_{\mathcal{J}_0'} A_{\mathcal{J}_0'}^T)^{-1} A_{\mathcal{J}_0'}.$$

XIII. Вычислить вектор

$$h^1 = -(I - H_{\mathcal{J}_0'}) \nabla f_0(x^0).$$

XIV. Положить $k = 0$ и перейти к шагу XVII.

Основной цикл. XV. Вычислить вектор $(I - H_{\mathcal{J}_0'}) \times \nabla f_0(x^k)$.

Если $(I - H_{\mathcal{J}_0'}) \nabla f_0(x^k) = 0$, то положить $x^0 = x^k$ и перейти к шагу II; иначе перейти к шагу XVI.

XVI. Вычислить вектор

$$h^{k+1} = -(I - H_{\mathcal{J}_0'}) \nabla f_0(x^k) + \frac{\| (I - H_{\mathcal{J}_0'}) \nabla f_0(x^k) \|^2}{\| (I - H_{\mathcal{J}_0'}) \nabla f_0(x^{k-1}) \|^2} h^k.$$

XVII. Вычислить

$$\rho_{k+1} = -(\nabla f_0(x^k),\ h^{k+1})/(h^{k+1},\ Ch^{k+1}).$$

XVIII. Найти подмножество $\mathcal{J}_0'' \subseteq \mathcal{J}_0'$ такое, что для всех $i \in \mathcal{J}_0''$ выполняется неравенство $(a^i,\ h^{k+1}) > 0$.

XIX. Вычислить

$$\bar{\rho}_{k+1} = \min_{i \in \mathcal{J}_0''} \frac{b_i - (a^i,\ x^k)}{(a^i,\ h^{k+1})}.$$

XX. Если $\rho_{k+1} < \bar{\rho}_{k+1}$, то перейти к шагу XXI; иначе положить $x^{k+1} = x^k + \bar{\rho}_{k+1} h^{k+1}$, $x^0 = x^{k+1}$ и перейти к шагу II.

XXI. Положить $x^{k+1} = x^k + \rho_{k+1} h^{k+1}$, $k = k+1$ и перейти к шагу XV.

XXII. Положить $\mathcal{J}_0' = \mathcal{J}_0$ и перейти к шагу XI.

Теорема 2. *Если выполнены предположения 2, то алгоритм 2 решает задачу 2 за конечное число итераций, при этом либо получается точка x^*, минимизирующая квадратичную функцию*

$$f_0(x) \triangleq \frac{1}{2}(Cx,\ x) + (d,\ x)$$

на множестве X, либо устанавливается тот факт, что $f_0(x)$ — неограничена снизу на множестве X.

Замечание 2. В случае вырожденности матрицы C может возникнуть ситуация, когда на шаге XVII алгоритма 2 будет выполняться $(\nabla f_0\ (x^k),\ h^{k+1}) \neq 0$, но $(h^{k+1},\ Ch^{k+1}) = 0$, что влечет $\rho_{k+1} = \infty$. Тогда, если $\bar{\rho}_{k+1} < \infty$, то осуществляется переход к шагу XX. Если же $\bar{\rho}_{k+1}$ неограничено, т. е. на шаге XVIII будет выполняться

$$(a^i,\ h^{k+1}) \leqslant 0 \text{ для всех } i \in \mathcal{J}_0',$$

то это означает, что задача 2 не имеет решения, так как нижняя грань функции $f_0(x)$ на множестве X равна $-\infty$.

Замечание 2'. Приведенные в пунктах 1 и 2 алгоритмы содержат, по существу, одну сложную вычислительную операцию: проектирование градиента на подпространство, т. е. вычисление вектора $(I - H_{\mathcal{J}(x)})\ \nabla f_0(x)$, что требует нахождения обратной матрицы $(A_{\mathcal{J}(x)} A^T_{\mathcal{J}(x)})^{-1}$.

Чтобы избежать вычисления обратной матрицы, можно вначале найти решение $\bar{u}^0$ задачи безусловной минимизации:

$$\text{найти } \arg\min_u \frac{1}{2} \|\nabla f_0(x) + A^T_{\mathcal{J}(x)} u\|^2. \tag{5.108}$$

(Задачу (5.108) можно решить, например, с помощью метода сопряженных направлений). Кроме того, решением задачи (5.108) является вектор

$$\bar{u}^0 = -(A_{\mathcal{J}(x)} A^T_{\mathcal{J}(x)})^{-1} A_{\mathcal{J}(x)} \nabla f_0(x).$$

Зная $\bar{u}_0$, можно найти вектор $(I - H_{\mathcal{J}(x)}) \nabla f_0(x)$ по формуле

$$(I - H_{\mathcal{J}(x)}) \nabla f_0(x) = \nabla f_0(x) + A^T_{\mathcal{J}(x)} \bar{u}^0.$$

3. Метод сопряженных градиентов для задачи квадратичного программирования с простыми ограничениями

Задача 3. Найти $\arg\min\limits_{x \in X} \left[\frac{1}{2}(Cx, x) + (d, x)\right]$,

где

$$X \triangleq \{x \mid x_i \geqslant 0, \quad i \in \mathcal{J}^+\};$$

$\mathcal{J}^+$ — некоторое подмножество множества $\{1, 2, \ldots, n\}$; $d \in R^n$.

Предположение 3. Матрица C положительно-определенная.

Определение 3. Определим множество $\mathcal{J}(x)$ и функцию $f_0(x)$

$$\mathcal{J}(x) \triangleq \{i \mid x_i = 0, \quad i \in \mathcal{J}^+\};$$

$$f_0(x) \triangleq \frac{1}{2}(Cx, x) + (d, x).$$

Алгоритм 3

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Найти множество $\mathcal{J}(x^0)$.

III. Вычислить градиент

$$\nabla f_0(x^0) \triangleq \left(\frac{\partial f_0(x^0)}{\partial x_1}, \ldots, \frac{\partial f_0(x^0)}{\partial x_n}\right)^T.$$

IV. Если

$$\frac{\partial f_0(x^0)}{\partial x_i} = 0, \quad \forall i \notin \mathcal{J}(x^0),$$

то перейти к шагу V; иначе перейти к шагу XVI.

V. Если
$$\frac{\partial f_0(x^0)}{\partial x_i} \geqslant 0, \quad \forall i \in \mathcal{J}(x^0),$$

то положить $x^* = x^0$ и прекратить вычисления (в этом случае находят оптимальное решение x^* задачи 3); иначе положить

$$\mathcal{J}'(x^0) = \left\{i \mid \frac{\partial f_0(x^0)}{\partial x_i} \geqslant 0, \quad i \in \mathcal{J}(x^0)\right\}$$

и перейти к шагу VI.

Примечание. На последующих шагах применяется метод сопряженных градиентов для минимизации функции $f_0(x)$, где в качестве переменных берут только x_i, $i \notin \mathcal{J}'(x^0)$, а все x_i, $i \in \mathcal{J}'(x^0)$ приравниваются к нулю.

VI. Положить $h^1 = -\nabla f_0(x^0)$.

VII. Положить $k = 0$ и перейти к шагу X.

Основной цикл. VIII. Вычислить

$$\nabla f_0(x^k) \triangleq \left(\frac{\partial f_0(x^k)}{\partial x_1}, \ldots, \frac{\partial f_0(x^k)}{\partial x_n}\right)^T.$$

Если $\frac{\partial f_0(x^k)}{\partial x_i} = 0$, $\forall i \notin \mathcal{J}'(x^0)$, то положить $x^0 = x^k$ и перейти к шагу II; иначе перейти к шагу IX.

IX. Вычислить вектор

$$h^{k+1} = -\nabla f_0(x^k) + \frac{\|\nabla f_0(x^k)\|^2}{\|\nabla f_0(x^{k-1})\|^2} h^k.$$

X. Вычислить

$$\rho_{k+1} = -(\nabla f_0(x^k),\ h^{k+1})/(h^{k+1},\ Ch^{k+1}).$$

XI. Определить $\bar{\mathcal{J}}(x^0)$ — множество всех $i \notin \mathcal{J}'(x^0)$, для которых $h_i^{k+1} < 0$.

XII. Вычислить

$$\bar{\rho}_{k+1} = \min_{i \in \bar{\mathcal{J}}(x^0)} (-x_i^k/h_i^{k+1}).$$

XIII. Если $\rho_{k+1} < \bar{\rho}_{k+1}$, то положить

$$x_i^{k+1} = x_i^k + \rho_{k+1} h_i^{k+1}, \quad i \notin \mathcal{J}'(x^0);$$

$$x_i^{k+1} = x_i^k = 0, \quad i \in \mathcal{J}'(x^0),$$

и перейти к шагу XIV; иначе положить

$$x_i^{k+1} = x_i^k + \bar{\rho}_{k+1} h_i^{k+1}, \quad i \notin \mathcal{J}'(x^0);$$

$$x_i^{k+1} = x_i^k = 0, \quad i \in \mathcal{J}'(x^0),$$

и перейти к шагу XV.

XIV. Положить $k = k + 1$ и перейти к шагу VIII.

XV. Положить $x^0 = x^{k+1}$ и перейти к шагу II.

XVI. Положить $\mathcal{J}'(x^0) = \mathcal{J}(x^0)$ и перейти к шагу VI.

Для алгоритма 3 имеют место теорема и замечание, аналогичные теореме 2 и замечанию 2.

4. Модификация метода сопряженных направлений для задач квадратичного программирования большой размерности

Задача 4. Найти $\arg\min_x f_0(x)$,

где

$$f_0(x) = \left(\frac{1}{2}(x,\ Cx) + (d,\ x)\right),$$

при ограничениях

$$(a^j,\ x) = b_j, \quad j = 1,\ \ldots,\ m; \tag{5.109}$$

$$v_i \leqslant x_i \leqslant w_i, \quad i = 1,\ \ldots,\ n, \tag{5.110}$$

где C — $n \times n$-матрица ($C \geqslant 0$); d — n-мерный вектор; a^j, $j = 1, \ldots, n$ — n-мерная строка, т. е. $a^j = (a_1^j, \ldots, a_n^j)$; b_j, $j = 1, \ldots, m$; v_i, w_i, $i = 1, \ldots, n$ — действительные числа.

Приводимая в этом пункте модификация метода сопряженных

направлений приспособлена для задач квадратичного программирования с большим числом переменных x_i, $i = 1, \ldots, n$, и относительно малым числом m ограничений вида (5.109). При отсутствии вычислительной погрешности приводимый алгоритм определяет те же точки, что и метод сопряженных направлений.

Отличием приводимой модификации от метода сопряженных направлений является то, что она требует меньшего количества вычислений, меньшего объема машинной памяти и позволяет устранять накапливающуюся погрешность машинных вычислений.

Алгоритм 4

Начало. I. Задать начальное приближение x^0, удовлетворяющее ограничениям-равенствам (5.109) и следующим неравенствам:

$$v_i < x_i^0 < w_i, \quad i = 1, \ldots, n.$$

II. Обозначить через A $m \times n$-матрицу, j-й строкой которой является a^j, $j = 1, \ldots, m$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить множества индексов $\mathcal{J}^+$, $\mathcal{J}^-$ по формулам

$$\mathcal{J}^+ = \{i \mid x_i^k \geqslant w_i, \quad i = 1, \ldots, n\};$$

$$\mathcal{J}^- = \{i \mid x_i^k \leqslant v_i, \quad i = 1, \ldots, n\}$$

и положить $\mathcal{J} = \mathcal{J}^+ \cup \mathcal{J}^-$.

V. Вычислить $\nabla f_0(x^k)$ — градиент функции цели в точке $x = x^k$

$$\nabla f_0(x^k) = Cx^k + d.$$

VI. Для произвольной матрицы B с m строками определить матрицу $\tilde{B}$, которая получается, если в B заменить все строки с номерами $j \in \mathcal{J}^+ \cup \mathcal{J}^-$ нулевыми строками, и определить матрицу B_+ (или B_-), строками которой являются строки матрицы B с номерами $j \in \mathcal{J}^+$ (соответственно, $j \in \mathcal{J}^-$).

VII. Для произвольного n-мерного вектор-столбца g определить операторы

$$Q_{\mathcal{J}} g = \tilde{g} - \tilde{A}^T (\tilde{A}\tilde{A}^T)^{-1} \tilde{A}\tilde{g}; \tag{5.111}$$

$$G_{\mathcal{J}} g = \begin{pmatrix} G_{\mathcal{J}}^+ g \\ G_{\mathcal{J}}^- g \end{pmatrix}, \tag{5.112}$$

где

$$G_{\mathcal{J}}^+ g = (A_+)^T (\tilde{A}\tilde{A}^T)^{-1} \tilde{A}\tilde{g} - g_+;$$

$$G_{\mathcal{J}}^- g = g_- - (A_-)^T (\tilde{A}\tilde{A}^T)^{-1} \tilde{A}\tilde{g}$$

(здесь $\tilde{A}^T$ — матрица, транспонированная к $\tilde{A}$).

VIII. Вычислить векторы $Q_{\mathcal{J}}\nabla f_0(x^k)$, $G_{\mathcal{J}}\nabla f_0(x^k)$ и положить $q = G_{\mathcal{J}}\nabla f_0(x^k)$.

IX. Если $Q_{\mathcal{J}}\nabla f_0(x^k) \neq 0$, то перейти к шагу XII; иначе перейти к шагу X.

X. Если при всех $i = 1, \ldots, n$ выполняется $q_i \geqslant 0$, то прекратить вычисления (в этом случае x^k является решением задачи 4); иначе перейти к шагу XI.

XI. Построить новые множества $\mathcal{J}^+$ и $\mathcal{J}^-$ путем удаления из них одного из номеров i таких, что $q_i < 0$ и перейти к шагу XII.

Отметим, что при изменении множеств $\mathcal{J}^+$ и $\mathcal{J}^-$ меняются также операторы $Q_{\mathcal{J}}$, $G_{\mathcal{J}}$.

На шагах XII—XVII осуществляется минимизация квадратичной функции $f_0(x)$ на подпространстве, задаваемом уравнениями

$$x_i^k = w_i, \quad i \in \mathcal{J}^+; \quad x_i^k = v_i, \quad i \in \mathcal{J}^-;$$

$$(a^j, x) = b_j, \quad j = 1, \ldots, m,$$

причем минимум $f_0(x)$ на указанном подпространстве находят не более чем за $n - m - \varkappa$ «малых» итераций, определяемых шагами XII—XVII, где $\varkappa$ — число элементов в множестве $\mathcal{J}^+ \cup \mathcal{J}^-$.

XII. Положить $s = 0$.

XIII. Положить $y^0 = x^k$, $h^0 = 0$.

XIV. Вычислить вектор

$$h^{s+1} = -Q_{\mathcal{J}}\nabla f_0(y^s) + (\|Q_{\mathcal{J}}\nabla f_0(y^s)\|^2/\|Q_{\mathcal{J}}\nabla f_0(y^{s-1})\|^2)\, h^s.$$

XV. Вычислить

$$\rho_{s+1} = -(\nabla f_0(y^s),\ h^{s+1})/(h^{s+1},\ Ch^{s+1}).$$

Если $\rho_{s+1} = 0$, то положить $x^{k+1} = y^s$ и перейти к шагу XVIII; иначе перейти к шагу XVI.

XVI. Вычислить величину $\bar{\rho}_{s+1}$ по правилу

$$\bar{\rho}_{s+1} = \min\left\{\min_{h_i^{s+1}>0} \frac{w_i - y_i^k}{h_i^{s+1}},\ \min_{h_i^{s+1}>0} \frac{y_i^k - v_i}{h_i^{s+1}}\right\}.$$

XVII. Если $\rho_{s+1} < \bar{\rho}_{s+1}$, то положить

$$y^{s+1} = y^s + \rho_{s+1} h^{s+1}, \quad s = s + 1,$$

и перейти к шагу XIV; если $\rho_{s+1} \geqslant \bar{\rho}_{s+1}$, то положить

$$x^{k+1} = y^s + \bar{\rho}_{s+1} h^{s+1}$$

и перейти к шагу XVIII.

XVIII. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 4. *Пусть итеративный процесс, задаваемый алгоритмом 4, реализован на ЭВМ и пусть y^s — точка, построенная на s-й*

итерации. Тогда выполняются следующие неравенства:

$$y_i^s - w_i \leqslant c_1 u; \quad v_i - y_i^s \leqslant c_1 u;$$

$$\| Ay^s - b \| \leqslant c_2 u (1 + \lambda^s) \| A \| + \| Ay^0 - b \|$$

(здесь c_1, c_2— константы; $u = 2^{1-\tau}$ — единичная ошибка округления (τ — число значащих разрядов мантисы); $b = (b_1, \ldots, b_m)$; λ — *константа, удовлетворяющая неравенствам*

$$\frac{1}{4} (\mathrm{cond}\, \tilde{C} + (\mathrm{cond}\, \tilde{C})^{-1} - 2) \leqslant \lambda \leqslant \frac{1}{2} (\mathrm{cond}\, \tilde{C} + (\mathrm{cond}\, \tilde{C})^{-1} - 2),$$

где $\tilde{C}$ — *матрица сужения функции* $f_0(x)$ *на гиперплоскость* (5.109); $\mathrm{cond}\, \tilde{C} \triangleq \| \tilde{C} \| \cdot \| \tilde{C}^{-1} \|$ — *число обусловленности матрицы* C).

Замечание 4. Так как погрешность выполнения равенства (5.109) может расти весьма быстро, то в [209] рекомендуется применять на некоторых итерациях по s следующую процедуру корректировки вычислительной погрешности:

$$y_{\text{кор}}^s = \theta Q y^s + (1 - \theta) y^0, \tag{5.113}$$

где Q — проектор $Q_{\mathcal{J}}$, определяемый по (5.111) при $\mathcal{J}^+ = \mathcal{J}^- = \varnothing$; θ — максимальное число из отрезка [0, 1] такое, что правая часть в (5.113) удовлетворяет ограничениям задачи.

5. Устойчивый алгоритм решения задач квадратичного программирования

Задача 5. Найти $\arg\min\limits_x \frac{1}{2} \| Gx - c \|^2$ при ограничениях

$$Ax = b, \quad x \geqslant 0,$$

где G — $l \times n$-матрица, состоящая из столбцов g^j, $j = 1, 2, \ldots, n$; c — l-вектор; A — $m \times n$-матрица, состоящая из столбцов a^j, $j = 1, \ldots, n$; b — m-вектор.

В приведенном ниже алгоритме на каждой итерации требуется решать систему линейных уравнений. Алгоритм за конечное число итераций приводит к решению задачи 5 либо устанавливает отсутствие допустимых решений задачи 5 и что важно, может применяться и в (почти) вырожденном случае задачи 5.

Алгоритм 5

Начало. I. Найти множество индексов $\mathcal{J} \subseteq \{1, 2, \ldots, n\}$ такое, что начальная матрица $\begin{pmatrix} B_{\mathcal{J}} \\ Q_{\mathcal{J}} \end{pmatrix}$ (здесь и далее $B_{\mathcal{J}}$ — подматрица матрицы A, состоящая из столбцов a^j, $j \in \mathcal{J}$, матрицы A; $Q_{\mathcal{J}}$ — подматрица матрицы G, состоящая из столбцов g^j, $j \in \mathcal{J}$, матрицы G), имеет полный столбцовый ранг и система линейных уравнений

$$B_{\mathcal{J}} y = b;$$
$$B_{\mathcal{J}}^T u + Q_{\mathcal{J}}^T Q_{\mathcal{J}} y = Q_{\mathcal{J}}^T c \tag{5.114}$$

имеет решение (y, u), для которого $y > 0$.

Основной цикл. II. Если $y > 0$, то перейти к шагу III; иначе перейти к шагу VII.

III. Если для всех $j \in \{1, \ldots, n\} \setminus \mathcal{J}$ выполняется неравенство

$$(a^j, u) + (g^j, Q_{\mathcal{J}} y - c) \geqslant 0,$$

то вычислить оптимальное решение $x^* = (x_1^*, \ldots, x_n^*)$ задачи 5

$$x_j^* = \begin{cases} y_j, & \text{если } j \in \mathcal{J}, \\ 0, & \text{если } j \notin \mathcal{J}, \end{cases} \quad j = 1, \ldots, n,$$

и прекратить вычисления; иначе положить $\overline{\mathcal{J}} = \mathcal{J}$ и перейти к шагу IV.

IV. Найти индекс $k \in \{1, \ldots, n\} \setminus \overline{\mathcal{J}}$, для которого выполняется неравенство

$$(a^k, u) + (g^k, Q_{\mathcal{J}} y - c) < 0.$$

V. Положить $\mathcal{J} = \overline{\mathcal{J}} \cup \{k\}$.

VI. Найти вектор $\bar{y}$, координаты которого $\bar{y}_j$, $j \in \mathcal{J}$, вычисляются по правилу

$$\bar{y}_i = y_i \text{ при всех } i \in \overline{\mathcal{J}};$$
$$\bar{y}_k = 0$$

и перейти к шагу XI.

VII. Вычислить максимальное значение $\bar{\lambda} \in [0, 1]$, при котором

$$\bar{\lambda} y + (1 - \bar{\lambda}) \bar{y} \geqslant 0,$$

и положить

$$z_j = \bar{\lambda} y_j + (1 - \bar{\lambda}) \bar{y}_j, \quad j \in \mathcal{J}.$$

VIII. Положить $\overline{\mathcal{J}} = \mathcal{J}$.

IX. Найти множество $\tilde{\mathcal{J}}$, состоящее из тех индексов $k \in \overline{\mathcal{J}}$, для которых $z_k = 0$ и положить $\mathcal{J} = \overline{\mathcal{J}} \setminus \tilde{\mathcal{J}}$.

X. Вычислить вектор $\bar{y}$, координаты которого находятся по правилу

$$\bar{y}_j = z_j, \quad j \in \mathcal{J},$$

и перейти к шагу XI.

XI. В соответствии с множеством $\mathcal{J}$ составить подматрицу $B_{\mathcal{J}}$ матрицы A и подматрицу $Q_{\mathcal{J}}$ матрицы G.

XII. Найти решение (y, u) системы линейных уравнений

$$B_{\mathcal{J}} y = b;$$
$$B_{\mathcal{J}} u + Q_{\mathcal{J}}^T Q_{\mathcal{J}} y = Q_{\mathcal{J}}^T c \tag{5.115}$$

и перейти к шагу II.

Теорема 5. *Пусть существует множество индексов* $\mathcal{J} \subseteq \{1, \ldots \ldots, n\}$ *такое, что матрица* $\begin{pmatrix} B_{\mathcal{J}} \\ Q_{\mathcal{J}} \end{pmatrix}$ *имеет полный столбцовый ранг и система (5.114) имеет решение* (y, u), *для которого* $y > 0$, *и пусть каждая матрица* $B_{\mathcal{J}}$, *получающаяся в результате применения алгоритма, имеет полный строчный ранг. Тогда за конечное число итераций алгоритма 5 вычисляется оптимальное решение* x^* *задачи 5.*

Замечание 5. Для получения начальной матрицы $\begin{pmatrix} B_{\mathcal{J}} \\ Q_{\mathcal{J}} \end{pmatrix}$, описанной на шаге I алгоритма 5, необходимо предварительно получить решение $(\tilde{t}^*, \tilde{x}^*)$ следующей задачи квадратичного программирования:

$$\text{найти } \arg\min_{(\tilde{t}, \tilde{x})} \left\| (-b \vdots A) \begin{pmatrix} \tilde{t} \\ \tilde{x} \end{pmatrix} \right\|^2 \tag{5.116}$$

при ограничениях

$$\tilde{t} = 1, \quad (\tilde{t}, \tilde{x}) \geqslant 0. \tag{5.117}$$

Если выполняется неравенство

$$\left\| (-b \vdots A) \begin{pmatrix} \tilde{t}^* \\ \tilde{x}^* \end{pmatrix} \right\|^2 > 0,$$

то задача 5 не имеет допустимого решения;
если

$$\left\| (-b \vdots A) \begin{pmatrix} \tilde{t}^* \\ \tilde{x}^* \end{pmatrix} \right\|^2 = 0,$$

то $\tilde{x}^*$ удовлетворяет условиям $\tilde{x}^* \geqslant 0$ и $A\tilde{x}^* = b$.

Если $\tilde{x}^* \neq 0$, то множество столбцов a^j, для которых $\tilde{x}_j^* > 0$, линейно-независимо и вместе с соответствующими столбцами g^j образует начальную матрицу $\begin{pmatrix} B_{\mathcal{J}} \\ Q_{\mathcal{J}} \end{pmatrix}$ алгоритма 5.

Отметим, что для решения задачи (5.116) — (5.117) можно использовать алгоритм 5 с начальной матрицей $\begin{pmatrix} B_{\mathcal{J}} \\ Q_{\mathcal{J}} \end{pmatrix}$, равной $\begin{pmatrix} 1 \\ -b \end{pmatrix}$.

Замечание 5'. Для решения систем линейных уравнений (5.114) и (5.115) в [524] предлагается следующий способ.

Пусть $\begin{pmatrix} B_{\mathcal{J}} \\ Q_{\mathcal{J}} \end{pmatrix}$ имеет размер $(m + l) \times r$. Тогда y и u определяются

с помощью решения треугольных линейных систем

$$S(b-u)=h; \quad Vy=d$$

для $(b-u)$ и y, где $d=W_Q^T c+Zh$; $h=S^{-T}b-Z^T W_Q^T c$; $W_B^T=ZS$; Z — $r\times m$-матрица с ортогональными столбцами и S — верхняя треугольная невырожденная $m\times m$-матрица, $\begin{pmatrix}B_{\mathcal{I}}\\ Q_{\mathcal{I}}\end{pmatrix}=WV$, W — $(m+l)\times r$-матрица с ортогональными столбцами, V — верхняя треугольная невырожденная $r\times r$-матрица, $W=\begin{pmatrix}W_B\\ W_Q\end{pmatrix}$, W_B — имеет m строк, W_Q — имеет l строк.

Библиографические указания. При написании параграфа использовались работы [209, 320, 523, 524]. Дополнительные сведения о методах решения задач квадратичного программирования можно найти в работах [35, 310, 311, 437, 461, 467, 473, 507, 546].

Глава 6

СПЕЦИАЛЬНЫЕ МЕТОДЫ РЕШЕНИЯ МИНИМАКСНЫХ ЗАДАЧ И МЕТОДЫ ОТЫСКАНИЯ СЕДЛОВЫХ ТОЧЕК

6.1. Методы последовательных приближений решения дискретных минимаксных задач

1. Минимаксная задача с ограничениями простой структуры

Задача 1. Найти $\arg\min_{x\in X}\max_{i\in\mathcal{I}}\varphi_i(x)$ для заданных функций $\varphi_i: \boldsymbol{R}^n\to\boldsymbol{R}^1$, $i\in\mathcal{I}$, заданного множества индексов $\mathcal{I}$ и заданного множества ограничений $X\subset\boldsymbol{R}^n$.

Предположения 1. (*i*) — функции φ_i, $i\in\mathcal{I}$ — непрерывно дифференцируемы; (*ii*) — множество X — выпукло и замкнуто.

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0\in X$.

II. Выбрать константы $\varepsilon_0>0$ и $\alpha_0>0$.

III. Положить $k=0$.

Основной цикл. IV. Найти множество точек

$$X^k=\{x\mid \|x-x^k\|_1\leqslant 1,\quad x\in X\},$$

где

$$\|y\|_1=\max_{i\in[1:n]}|y_i|.$$

V. Вычислить множество индексов

$$\mathcal{K}_0(x^k)=\{i\mid \varphi_i(x^k)=\max_{i\in\mathcal{I}}\varphi_i(x^k),\quad i\in\mathcal{I}\}.$$

VI. Вычислить

$$\psi(x^k) = \min_{x \in X^k} \max_{i \in \mathcal{K}_0(x^k)} (\nabla\varphi_i(x^k), \quad x - x^k).$$

VII. Если $\psi(x^k) = 0$, то положить $x^* = x^k$ и прекратить вычисления, если $\psi(x^k) < 0$, то перейти к шагу VIII.

VIII. Положить $j = 0$.

IX. Положить $\varepsilon = \varepsilon_j$.

X. Вычислить множество индексов

$$\mathcal{K}_\varepsilon(x^k) = \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \quad i \in \mathcal{I}\}.$$

XI. Вычислить

$$\psi_\varepsilon(x^k) = \min_{x \in X^k} \max_{i \in \mathcal{K}_\varepsilon(x^k)} (\nabla\varphi_i(x^k), \ x - x^k).$$

XII. Если выполняется неравенство $\psi_\varepsilon(x^k) \leqslant -\alpha_0\varepsilon/\varepsilon_0$, то перейти к шагу XIII; иначе положить $\varepsilon_{j+1} = \varepsilon_j/2$, $j = j + 1$ и перейти к шагу IX.

(выход из цикла, определяемого шагами IX—XII, будет осуществляться за конечное число итераций при каждом $k = 0, 1, \ldots$).

XIII. Вычислить точку y^k, удовлетворяющую условию

$$\psi_\varepsilon(x^k) = \max_{i \in \mathcal{K}_\varepsilon(x^k)} (\nabla\varphi_i(x^k), \ y^k - x^k).$$

XIV. Вычислить шаговый множитель ρ_k из условия

$$\max_{i \in \mathcal{I}} \varphi_i(x^k + \rho_k(y^k - x^k)) = \min_{\rho \in [0,1]} \max_{i \in \mathcal{I}} \varphi_i(x^k + \rho(y^k - x^k))$$

XV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k(y^k - x^k).$$

XVI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 1. *Если выполнены предположения 1 и начальное приближение x^0 в алгоритме 1 таково, что множество*

$$X(x^0) = \{x \mid \max_{i \in \mathcal{I}} \varphi_i(x) \leqslant \max_{i \in \mathcal{I}} \varphi_i(x^0), \ x \in X\}$$

ограничено, то любая предельная точка x^* бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 1, является стационарной точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X.

Замечание 1. Основная трудность в применении алгоритма 1 связана с вычислением значения $\psi_\varepsilon(x^k)$ на шаге XI. В [127] показано, что если X^k — строго выпуклое множество и начало координат не принадлежит выпуклой оболочке $L_\varepsilon(x^k)$, натянутой на векторы $\nabla\varphi_i(x^k)$, $i \in \mathcal{K}_\varepsilon(x^k)$, то задача вычисления $\psi_\varepsilon(x^k)$ сводится к более простой задаче максимизации на $L_\varepsilon(x^k)$ непрерывно дифференцируемой функции

$$\theta(z) \triangleq \min_{v \in X^k} (z, \ v - x^k),$$

градиент которой

$$\nabla\theta(z) = v(z) - x^k,$$

где $v(z) \in X^k$ удовлетворяет условию $\theta(z) = (z, v(z) - x^k)$.

2. Первый метод последовательных приближений решения минимаксной задачи с ограничениями типа неравенств

Задача 2. Найти $\arg\min_{x \in X} \max_{i \in \mathcal{I}} \varphi_i(x)$ для заданных функций $\varphi_i : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $i \in \mathcal{I}$, заданного множества индексов $\mathcal{I}$ и множества

$$X \triangleq \{x \mid f_j(x) \leqslant 0, \quad j \in \mathcal{I}_1, \quad x \in \boldsymbol{R}^n\}.$$

Предположения 2. (*i*) — функции φ_i, $i \in \mathcal{I}$ — непрерывно дифференцируемы; (*ii*) — функции f_j, $j \in \mathcal{I}_1$ — выпуклы и непрерывно дифференцируемы; (*iii*) — выполняется условие Слейтера

$$\inf_{x \in \boldsymbol{R}^n} \max_{i \in \mathcal{I}_1} f_i(x) < 0.$$

В пунктах 2—4 приводятся методы последовательных приближений для нахождения стационарных точек x^* функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X.

Точка $x^* \in X$ называется *стационарной* точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X, если многогранник $\tilde{L}(x^*)$ содержит начало координат (здесь

$$\tilde{L}(x^*) = \operatorname{co} \hat{H}(x^*), \quad \hat{H}(x^*) = H(x^*) \cup H_1(x^*);$$

$$H(x^*) = \{\nabla\varphi_i(x^*), \quad i \in \mathcal{K}_0(x^*)\};$$

$$H_1(x^*) = \{\nabla f_j(x^*), \quad j \in Q_0(x^*)\};$$

$$\mathcal{K}_0(x^*) = \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^*) = \varphi_i(x^*), \quad i \in \mathcal{I}\};$$

$$Q_0(x^*) = \{j \mid f_j(x^*) = 0, \quad j \in \mathcal{I}_1\}).$$

Если $\max_{i \in \mathcal{I}} \varphi_i(x)$ выпукла на X, то стационарная точка x^* является точкой минимума.

Алгоритм 2

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Выбрать константы $\varepsilon_0 > 0$, $\mu_0 > 0$, $\alpha_0 > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Найти множества индексов

$$\mathcal{K}_0(x^k) = \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) = \varphi_i(x^k), \quad i \in \mathcal{I}\};$$

$$Q_0(x^k) = \{j \mid f_j(x^k) = 0, \quad j \in \mathcal{I}_1\}.$$

V. Вычислить множество векторов

$$\breve{H}_{00}(x^k) = H_0(x^k) \cup H_0'(x^k),$$

где

$$H_0(x^k) = \{\nabla\varphi_i(x^k), \quad i \in \mathcal{K}_0(x^k)\};$$

$$H_0'(x^k) = \{\nabla f_j(x^k), \quad j \in Q_0(x^k)\}.$$

VI. Используя алгоритм 1′ (см. § 3.12), определить, принадлежит ли начало координат многограннику $\tilde{L}_{00}(x^k)$, который является выпуклой оболочкой, натянутой на множество векторов $\tilde{H}_{00}(x^k)$. Если начало координат принадлежит многограннику $\tilde{L}_{00}(x^k)$, то положить $x^* = x^k$ и прекратить вычисления; иначе перейти к шагу VII.

VII. Положить $s = 0$.

VIII. Положить $\varepsilon = \varepsilon_s$, $\mu = \mu_s$, $\alpha = \alpha_s$.

IX. Найти множества индексов

$$\mathcal{K}_\varepsilon(x^k) = \{i \mid \max_{i \in \mathcal{J}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \ i \in \mathcal{J}\};$$

$$Q_\varepsilon(x^k) = \{j \mid -\mu \leqslant f_j(x^k) \leqslant 0, \quad j \in \mathcal{J}_1\}.$$

X. Вычислить множество векторов

$$\tilde{H}_{\varepsilon\mu} = H_\varepsilon(x^k) \cup H_\mu'(x^k),$$

где множества

$$H_\varepsilon(x^k) = \{\nabla\varphi_i(x^k), \quad i \in \mathcal{K}_\varepsilon(x^k)\};$$

$$H_\mu'(x^k) = \{\nabla f_j(x^k), \quad j \in Q_\varepsilon(x^k)\}.$$

XI. Используя алгоритм 1″ (см. § 3.12), вычислить точку $z_{\varepsilon\mu}$ — ближайшую к началу координат точку многогранника $\tilde{L}_{\varepsilon\mu}(x^k)$, который является выпуклой оболочкой, натянутой на множество векторов $\tilde{H}_{\varepsilon\mu}(x^k)$.

XII. Если выполняется неравенство $\| z_{\varepsilon\mu} \| \geqslant \alpha$, то перейти к шагу XIII; иначе положить $\varepsilon_{s+1} = \varepsilon_s/2$, $\mu_{s+1} = \mu_s/2$, $\alpha_{s+1} = \alpha_s/2$, $s = s + 1$ и перейти к шагу VIII.

XIII. Вычислить вектор $h_{\varepsilon\mu}(x^k)$ — направление (ε, μ)-квазинаискорейшего спуска функции $\max_{i \in \mathcal{J}} \varphi_i(x)$ в точке x^k по формуле

$$h_{\varepsilon\mu}(x^k) = -(1/\| z_{\varepsilon\mu} \|)\, z_{\varepsilon\mu}.$$

XIV. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям

$$\max_{i \in \mathcal{J}} \varphi_i(x^k + \rho_k h_{\varepsilon\mu}(x^k)) = \min_{\rho \geqslant 0} \max_{i \in \mathcal{J}} \varphi_i(x^k + \rho h_{\varepsilon\mu}(x^k));$$

$$x^k + \rho_k h_{\varepsilon\mu}(x^k) \in X.$$

XV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h_{\varepsilon\mu}(x^k).$$

XVI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Если выполнены предположения 2 и начальное приближение $x^0 \in X$ таково, что множество*

$$X^0 \triangleq \{x \mid \max_{i \in \mathcal{I}} \varphi_i(x) \leqslant \max_{i \in \mathcal{I}} \varphi_i(x^0), \quad x \in X\}$$

ограничено, то любая предельная точка бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 2, является стационарной точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X.

3. Второй метод последовательных приближений решения минимаксной задачи с ограничениями типа неравенств

Алгоритм 3

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Выбрать константы $\varepsilon > 0$, $\mu > 0$

III. Положить $k = 0$.

Основной цикл. IV. Найти множества индексов

$$\mathcal{K}_\varepsilon(x^k) = \{i \mid \max_{i \in \mathcal{I}} \varphi_i(x^k) - \varphi_i(x^k) \leqslant \varepsilon, \quad i \in \mathcal{I}\};$$

$$Q_\varepsilon(x^k) = \{j \mid -\mu \leqslant f_j(x^k) \leqslant 0, \quad j \in \mathcal{I}_1\}.$$

V. Вычислить множество векторов

$$\tilde{H}_{\varepsilon\mu} = H_\varepsilon(x^k) \cup H'_\mu(x^k),$$

где множества

$$H_\varepsilon(x^k) = \{\nabla\varphi_i(x^k), \quad i \in \mathcal{K}_\varepsilon(x^k)\};$$

$$H'_\mu(x^k) = \{\nabla f_j(x^k), \quad j \in Q_\varepsilon(x^k)\}.$$

VI. Используя алгоритм 1′ (см. § 3.12), определить, принадлежит ли начало координат многограннику $\tilde{L}_{\varepsilon\mu}(x^k)$, который является выпуклой оболочкой, натянутой на множество векторов $\tilde{H}_{\varepsilon\mu}(x^k)$.

Если начало координат принадлежит многограннику $\tilde{L}_{\varepsilon\mu}(x^k)$, то положить $\overset{*}{x}_{\varepsilon\mu} = x^k$ и прекратить вычисления; иначе перейти к шагу VII.

VII. Используя алгоритм 1″ (см. § 3.12), вычислить точку $z^k_{\varepsilon\mu}$ — ближайшую к началу координат точку многогранника $\tilde{L}_{\varepsilon\mu}(x^k)$. Положить $\delta^k_{\varepsilon\mu} = \| z^k_{\varepsilon\mu} \|$.

VIII. Вычислить вектор $h_{\varepsilon\mu}(x^k)$ — направление (ε, μ)-наискорейшего спуска функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ в точке x^k по формуле

$$h_{\varepsilon\mu}(x^k) = -(1/\delta^k_{\varepsilon\mu})\, z^k_{\varepsilon\mu}.$$

IX. Вычислить шаговый множитель ρ_k, удовлетворяющий условиям

$$\max_{i \in \mathcal{I}} \varphi_i (x^k + \rho_k h_{\varepsilon\mu}(x^k)) = \min_{\rho \geq 0} \max_{i \in \mathcal{I}} \varphi_i (x^k + \rho h_{\varepsilon\mu}(x^k));$$

$$x^k + \rho_k h_{\varepsilon\mu}(x^k) \in X.$$

X. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h_{\varepsilon\mu}(x^k).$$

XI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 3. *Если выполнены все условия теоремы 2, то любая предельная точка бесконечной последовательности $\{x^k\}_{k=0}^{\infty}$, порожденной алгоритмом 3, является (ε, μ)-квазистационарной точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X.*

(Точка $x^*_{\varepsilon\mu} \in X$ называется (ε, μ)-квазистационарной точкой функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X, если $0 \in \tilde{L}_{\varepsilon\mu}(x^*_{\varepsilon\mu})$).

4. Модификация второго метода последовательных приближений

Алгоритм 3 для нахождения (ε, μ)-квазистационарных точек используется в приводимом ниже алгоритме для отыскания стационарных точек функции $\max_{i \in \mathcal{I}} \varphi_i(x)$ на множестве X.

Алгоритм 4

Начало. I. Выбрать начальное приближение $\bar{x}^0 \in X$.

II. Выбрать константы $\varepsilon_0 > 0$, $\mu_0 > 0$, $\alpha_0 > 0$.

III. Положить $s = 0$.

Основной цикл. IV. Найти множества индексов

$$\mathcal{K}_0(\bar{x}^s) = \{i \mid \max_{i \in \mathcal{I}} \varphi_i(\bar{x}^s) = \varphi_i(\bar{x}^s) \quad i \in \mathcal{I}\};$$

$$Q_0(\bar{x}^s) = \{j \mid f_j(\bar{x}^s) = 0, \quad j \in \mathcal{I}_1\}.$$

V. Вычислить множество векторов

$$\tilde{H}_{00}(\bar{x}^s) = H_0(\bar{x}^s) \cup H_0'(\bar{x}^s),$$

где множества

$$H_0(\bar{x}^s) = \{\nabla\varphi_i(\bar{x}^s), \quad i \in \mathcal{K}_0(\bar{x}^s)\};$$

$$H_0'(\bar{x}^s) = \{\nabla f_j(\bar{x}^s), \quad j \in Q_0(\bar{x}^s)\}.$$

VI. Используя алгоритм 1' (см. § 3.12), определить, принадлежит ли начало координат многограннику $L_{00}(\bar{x}^s)$, который является выпуклой оболочкой, натянутой на множество векторов $\tilde{H}_{00}(\bar{x}^s)$. Если начало координат принадлежит многограннику

$\tilde{L}_{00}(\bar{x}^s)$, то положить $x^* = \bar{x}^s$ и прекратить вычисления; иначе перейти к шагу VII.

VII. Используя алгоритм 3, в котором положить

$$\mu = \mu_s, \quad \varepsilon = \varepsilon_s, \quad x^0 = \bar{x}^s,$$

найти первый индекс k, при котором выполняется неравенство $\delta^k_{\varepsilon\mu} \leqslant \alpha_s$ и положить $\bar{x}^{s+1} = x^k$.

VIII. Положить $\varepsilon_{s+1} = \varepsilon_s/2$, $\mu_{s+1} = \mu_s/2$, $\alpha_{s+1} = \alpha_s/2$.

IX. Положить $s = s + 1$ и перейти к шагу IV.

Теорема 4. *Если выполнены все условия теоремы 2, то любая предельная точка бесконечной последовательности* $\{\bar{x}^s\}_{s=0}^{\infty}$, *порожденной алгоритмом 4, является стационарной точкой функции* $\max_{i\in\mathcal{I}} \varphi_i(x)$ *на множестве* X.

Библиографические указания. При написании параграфа использовались работы [254, 127, 428, 480, 481, 529].

6.2. Обобщенный беспараметрический метод внешней точки решения дискретных минимаксных задач

Задача 0. Найти $\arg\min_{x\in X} \max_{i\in[1,m]} f_i(x)$, где

$$X \triangleq \{x \mid g_j(x) \leqslant 0, \quad j \in \mathcal{J}, \quad h_l(x) = 0, \quad l \in L, \quad x \in R^n\};$$

$$g_j : R^n \to R^1, \quad j \in \mathcal{J}, \quad h_l : R^n \to R^1, \quad l \in L;$$

$\mathcal{J}$ и L — конечные множества индексов.

Предположения 0. (*i*) — функции $f_i(x)$, $i = 1, \ldots, m$, $g_j(x)$, $j \in \mathcal{J}$, $h_l(x)$, $l \in L$ — непрерывно дифференцируемые в R^n; (*ii*) — существует решение x^* минимаксной задачи 0.

Обозначим

$$f_0(x) = \max_{i\in[1,m]} f_i(x); \quad \mathcal{J}(x) = \{j \mid g_j(x) > 0, \quad j \in \mathcal{J}\}.$$

1. Основной алгоритм

На k-й итерации алгоритма вычисляется точка x^k безусловного минимума некоторой вспомогательной функции и оценка снизу α_k для значения $f_0(x^*)$ такие, что

$$f_0(x^k) \to f_0(x^*) \quad \text{при} \quad k \to \infty;$$

$$\alpha_k \to f_0(x^*) \quad \text{при} \quad k \to \infty;$$

$$\alpha_{k+1} \geqslant \alpha_k, \quad k = 0, 1, \ldots; \quad f_0(x^*) \geqslant \alpha_k, \quad k = 0, 1, \ldots .$$

Для начала работы алгоритма необходима нижняя оценка α_0 значения $f_0(x^*)$.

Алгоритм 1

Начало. I. Выбрать константу $\alpha_0 \leqslant f_0(x^*)$ (т. е. нижнюю оценку α_0 оптимального значения $f_0(x^*)$ задачи 0).

II. Задать произвольные веса w_j, $j \in \mathcal{I}$ и v_l, $l \in L$.

III. Для фиксированного числа α определить функцию

$$\varphi(x, \alpha) = \sum_{i \in \mathcal{K}(x)} [f_i(x) - \alpha]^2 + \sum_{i \in \mathcal{I}(x)} w_j (g_j(x))^2 + \sum_{l \in L} v_l (h_l(x))^2,$$

где $\mathcal{K}(x) = \{i \mid f_i(x) > \alpha,\ i = 1, \ldots, m\}$.

IV. Положить $k = 0$.

Основной цикл. V. Найти точку x^k, удовлетворяющую условию

$$\varphi(x^k, \alpha_k) = \min_{x \in \mathbf{R}^n} \varphi(x, \alpha_k),$$

т. е. решить задачу безусловной минимизации дифференцируемой по x функции $\varphi(x, \alpha_k)$.

VI. Вычислить

$$\alpha_{k+1} = \alpha_k + [\varphi(x^k, \alpha_k)/m]^{1/2}. \qquad (6.1)$$

VII. Положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Пусть выполняются предположения 0 и пусть: (iii) — $f_0(\bar{x}(u))$ — непрерывная функция аргумента и в точке $u = 0$, где $\bar{x}(u)$ является решением следующей задачи:*

$$\text{найти } \arg\min_x \max_{i \in [1, m]} f_i(x)$$

при ограничениях

$$g_j(x) \leqslant u_j, \quad j \in \mathcal{I};$$
$$h_l(x) = u_l, \quad l \in L,$$

здесь u_j, $j \in \mathcal{I}$, u_l, $l \in L$ — компоненты вектора u.

Тогда алгоритм 1 порождает последовательности $\{x^k\}_{k=0}^{\infty}$, $\{\alpha_k\}_{k=0}^{\infty}$ такие, что

$$\alpha_k \to f_0(x^*) \quad \text{при} \quad k \to \infty;$$
$$f_0(x^k) \to f_0(x^*) \quad \text{при} \quad k \to \infty;$$
$$\varphi(x^k, \alpha_k) \to 0 \quad \text{при} \quad k \to \infty.$$

2. Ускоренный вариант алгоритма

В ускоренном варианте алгоритма на некоторых итерациях вычисление величины α_{k+1} проводится не по (6.1), а по формуле

$$\alpha'_{k+1} = \alpha'_k + \varphi(x^k, \alpha_k) / \sum_{i \in \mathcal{K}(x^k)} (f_i(x^k) - \alpha'_k),$$

при этом справедливо неравенство $\alpha'_{k+1} \geqslant \alpha_{k+1}$, что обеспечивает ускорение сходимости $\{\alpha'_k\}_{k=1}^{\infty}$ к $f_0(x^*)$ по крайней мере при α'_k близких к $f_0(x^*)$.

Алгоритм 2

Начало. I—III. Шаги I—III такие, как и в алгоритме 1.

IV. Выбрать верхнюю оценку $\bar{\alpha}_0$ оптимального значения $f_0(x^*)$ задачи 0.

V. Выбрать константу $\varepsilon_f > 0$, характеризующую точность вычисления $f_0(x^*)$, константу ε_φ, характеризующую близость значений функции $\varphi(x, \alpha)$ к нулю.

VI. Положить $\tau_1 = \alpha_0$.

VII. Положить $k = 1$.

Основной цикл. VIII. Найти точку x^k, удовлетворяющую условию

$$\varphi(x^k, \tau_k) = \min_{x \in R^n} \varphi(x, \tau_k).$$

IX. Вычислить значения t_1 и t_2, соответственно

$$t_1 = \tau_k + [\varphi(x^k, \tau_k)/m]^{1/2};$$

$$t_2 = \tau_k + \varphi(x^k, \tau_k) / \sum_{i \in \mathcal{K}(x^k)} (f_i(x^k) - \tau_k).$$

X. Если $t_2 \leqslant \bar{\alpha}_0$, то положить $\tau_{k+1} = t_2$ и перейти к шагу XI; иначе положить $\tau_{k+1} = t_1$ и перейти к шагу XI.

XI. Положить $\tau_0 = \tau_{k+1} - \tau_k$; $\alpha_0 = t_1$; $s = \varphi(x^k, \tau_k)$.

XII. Положить $k = k + 1$ и перейти к шагу XIII.

XIII. Найти точку x^k, удовлетворяющую условию

$$\varphi(x^k, \tau_k) = \min_{x \in R^n} \varphi(x, \tau_k).$$

XIV. Если $\bar{\alpha}_0 - \alpha_0 \leqslant \varepsilon_f$ или $\tau_0 \leqslant \varepsilon_f$, то прекратить вычисления (в этом случае находят приближенное значение для $f_0(x^*)$, которое можно вычислить, например, по правилу $f_0(x^*) \cong (\bar{\alpha}_0 + \alpha_0)/2$); иначе перейти к шагу XV.

XV. Если $\varphi(x^k, \tau_k) > 0$, то перейти к шагу IX; если $s < \varepsilon_\varphi$, то прекратить вычисления (в этом случае также находят приближенное значение для $f_0(x^*)$); иначе положить $\bar{\alpha}_0 = \tau_k$, $s = 0$, $\tau_k = \alpha_0$ и перейти к шагу XIII.

Замечание 2. Алгоритм 2 основан на том, что неравенство $\tau \geqslant f_0(x^*)$ справедливо тогда и только тогда, когда $\inf_{x \in R^n} \varphi(x, \tau) = 0$

Библиографические указания. Параграф основан на работе [460].

6.3. Метод второго порядка решения задач дискретного минимакса

Задача 1. Найти $\arg \min_{x \in X} \max_{i \in \mathcal{K}} \varphi_i(x)$ для заданных функций $\varphi_i : R^n \to R^1$, $i \in \mathcal{K}$ и заданного множества

$$X = \{x \mid f_j(x) \leqslant 0, \quad j \in \mathcal{J}, \quad x \in R^n\},$$

где $\mathcal{K}$, $\mathcal{J}$ — конечные множества индексов.

Предположение 1. (*i*) — функции $\varphi_i(x)$, $i \in \mathcal{K}$ и $f_j(x)$, $j \in \mathcal{J}$ — дважды непрерывно дифференцируемые в $\boldsymbol{R}^n$.

В приводимом методе используется квадратичная аппроксимация $\varphi_i^k(x)$, $f_j^k(x)$ соответственно, функций $\varphi_i(x)$, $f_j(x)$ в точке $x = x^k$, что приводит к сверхлинейной скорости сходимости с показателем, равным $^3/_2$. На каждой итерации алгоритма требуется решать (точно) некоторую вспомогательную минимаксную задачу.

Алгоритм 1

Начало. I. Выбрать начальное приближение x^0, принадлежащее достаточно малой окрестности множества X.

II. Выбрать произвольные константы $0 < \varepsilon < 1$, $\delta > 0$, $\mu > 0$ (обычно $\varepsilon = {}^1/_2$, $\delta \in [10^{-3}, 10^{-1}]$, $\mu \in [10^{-3}, 10^{-1}]$).

III. Выбрать достаточно большую константу $\alpha > 0$.

IV. Положить $k = 0$.

Основной цикл. V. Вычислить $\varphi_i(x^k)$, $f_j(x^k)$, $\nabla\varphi_i(x^k)$, $\nabla f_j(x^k)$, $\nabla^2_{xx}\varphi_i(x^k)$, $\nabla^2_{xx}f_j(x^k)$ при всех $i \in \mathcal{K}$, $j \in \mathcal{J}$.

VI. Вычислить множества индексов

$$\mathcal{K}_k = \{i \mid \varphi_i(x^k) \geqslant \max_{i \in \mathcal{K}} \varphi_i(x^k) - \delta, \quad i \in \mathcal{K}\};$$

$$\mathcal{J}_k = \{j \mid f_j(x^k) \geqslant \max_{i \in \mathcal{J}} f_j(x^k) - \mu, \quad j \in \mathcal{J}\}.$$

VII. Определить функции $\varphi_i^k(x), f_j^k(x)$, $i \in \mathcal{K}$, $j \in \mathcal{J}$, соответственно

$$\varphi_i^k(x) = \varphi_i(x^k) + (\nabla\varphi_i(x^k),\ x - x^k) + \frac{1}{2}\nabla^2_{xx}\varphi_i(x^k)(x - x^k)^2;$$

$$f_j^k(x) = f_j(x^k) + (\nabla f_j(x^k),\ x - x^k) + \frac{1}{2}\nabla^2_{xx}f_j(x^k)(x - x^k)^2.$$

VIII. Найти решение $\bar{x}^k$ следующей вспомогательной задачи выпуклого программирования:

$$\text{найти } \arg\min_{x \in X_k} \max_{i \in \mathcal{K}_k} \varphi_i^k(x), \tag{6.2}$$

где множество

$$X_k = \{x \mid f_j^k(x) \leqslant 0, \quad j \in \mathcal{J}_k, \quad x \in \boldsymbol{R}^n\}.$$

IX. Положить $h^k = \bar{x}^k - x^k$. Если $\|h^k\| = 0$, то прекратить вычисления (в этом случае $\bar{x}^k$ — решение задачи 1); иначе перейти к шагу X.

X. Положить $\rho = 1$.

XI. Определить функции

$$\psi(x) = \max_{i \in \mathcal{K}} \varphi_i(x); \quad q(x) = \max\{0, \max_{i \in \mathcal{J}} f_j(x)\}.$$

XII. Если выполняется неравенство

$$\psi(x^k + \rho h^k) + \alpha q(x^k + \rho h^k) \leqslant \psi(x^k) + \alpha q(x^k) +$$
$$+ \varepsilon\rho [\max_{i \in \mathcal{K}_k} \varphi_i^k(\bar{x}^k) - \psi(x^k) - \alpha q(x^k)],$$

то положить $\rho_k = \rho$ и перейти к шагу XIV; иначе перейти к шагу XIII.

XIII. Положить $\rho = \rho/2$ и перейти к шагу XII.

XIV. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k h^k,$$

положить $k = k + 1$ и перейти к шагу V.

Теорема 1. *Пусть выполняется предположение 1 и, кроме того:* (*ii*) — *для любых* $x, y \in \boldsymbol{R}^n$, $i \in \mathcal{K}$, $j \in \mathcal{J}$, *справедливо*

$$\gamma_1 \| x \|^2 \leqslant (\nabla_{xx}^2 \varphi_i(y) x, x) \leqslant \gamma_2 \| x \|^2;$$
$$\gamma_1 \| x \|^2 \leqslant (\nabla_{xx}^2 f_j(y) x, x) \leqslant \gamma_2 \| x \|^2, \quad \gamma_2 \geqslant \gamma_1 > 0;$$

(*iii*) — *для задачи 1 и вспомогательных задач (6.2) выполняется условие Слейтера;* (*iv*) — *множители Лагранжа* $\beta_j(x^k)$, $j \in \mathcal{J}_k$, *вспомогательных задач (6.2), соответствующие ограничениям* $f_j^k(x) \leqslant 0$, *равномерно ограничены:*

$$\sum_{j \in \mathcal{J}_k} \beta_j(x^k) \leqslant \alpha; \quad \forall x^k \in Y_\alpha \triangleq \{x \mid \max_{i \in \mathcal{K}} \varphi_i(x) +$$
$$+ \alpha \max\{0, \max_{j \in \mathcal{J}} f_j(x)\} \leqslant \max_{i \in \mathcal{K}} \varphi_i(x^0) +$$
$$+ \alpha \max\{0, \max_{j \in \mathcal{J}} f_j(x^0)\}, \quad x \in \boldsymbol{R}^n\}.$$

Тогда для последовательности $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, справедливы утверждения:*

1) $\lim\limits_{k \to \infty} x^k = x^*$;

2) *если матрицы* $\nabla_{xx}^2 \varphi_i(x)$, $\nabla_{xx}^2 f_j(x)$ при

$$i \in \mathcal{K}^* \triangleq \{i \mid \varphi_i(x^*) = \max_{i \in \mathcal{K}} \varphi_i(x^*), \quad i \in \mathcal{K}\},$$
$$j \in \mathcal{J}^* \triangleq \{j \mid f_j(x^*) = \max_{j \in \mathcal{J}} f_j(x^*), \quad j \in \mathcal{J}\}$$

непрерывны в окрестности точки x^*, *то начиная с некоторой итерации* $\rho_k \equiv 1$; $\lim\limits_{k \to \infty} \| h^k \| / \| h^{k-1} \| = 0$;

3) *если, кроме того, матрицы* $\nabla_{xx}^2 \varphi_i(x)$, $\nabla_{xx}^2 f_j(x)$ *при* $i \in \mathcal{K}^*$, $j \in \mathcal{J}^*$, *удовлетворяют условию Липшица, то при* $k \to \infty$ $\| h^k \| \leqslant \leqslant \nu_1 \| h^{k-1} \|^{3/2}$,

где $\nu_1 > 0$ — *постоянная величина.*

Замечание 1. Поскольку константу α, обеспечивающую выполне-

ние условия (iv) теоремы 1, выбрать заранее довольно трудно, то в [280, 320] рекомендуется выбирать

$$\alpha = 2 \sum_{i \in \mathcal{I}_k} \beta_i (x^k)$$

всякий раз, когда (iv) не выполняется.

Замечание 1'. Достоинством приведенного в этом параграфе метода является его сходимость с любого начального приближения x^0, принадлежащего достаточно малой окрестности множества X. В то же время метод требует на каждой итерации решать вспомогательную задачу (6.2), для которой не существует в общем случае эффективных алгоритмов.

Библиографические указания. Параграф написан на основании работы [280].

6.4. Сеточный метод последовательных приближений решения непрерывных минимаксных задач

Задача 1. Найти $\arg \min_{x \in X} \max_{y \in Y} \varphi(x, y)$ для заданных функции $\varphi : \boldsymbol{R}^n \times \boldsymbol{R}^m \to \boldsymbol{R}^1$, ограниченного замкнутого множества $Y \subset \boldsymbol{R}^m$, выпуклого замкнутого множества $X \subset \boldsymbol{R}^n$.

Предположение 1. Функция $\varphi(x, y)$ непрерывна вместе с $\nabla_x \varphi(x, y)$ по совокупности переменных в $X' \times Y$, где X' — произвольное открытое множество, содержащее X.

Ниже описывается метод вычисления стационарных точек функции $\max_{y \in Y} \varphi(x, y)$ на множестве X, т. е. точек x^*, для которых

$$\inf_{x \in X} \max_{y \in \mathcal{K}(x^*)} (\nabla_x \varphi(x^*, y), x - x^*) = 0,$$

где

$$\mathcal{K}(x) = \{y \mid \varphi(x, y) = \max_{y \in Y} \varphi(x, y), \quad y \in Y\}.$$

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Положить $k = 0$.

Основной цикл. III. Построить сетку

$$Y_{N_k} = \{y^i \mid i \in [0 : N_k], \quad y^i \in Y\},$$

удовлетворяющую условиям теоремы 1.

IV. Определить функции $\varphi_i : \boldsymbol{R}^n \to \boldsymbol{R}^1$, $i \in [0 : N_k]$, по правилу

$$\varphi_i(x) \triangleq \varphi(x, y^i), \quad i \in [0 : N_k].$$

V. Используя алгоритмы § 6.1, найти хотя бы одну стационарную точку $\bar{x}^k \in X$ функции $\max_{i \in [0:N_k]} \varphi_i(x)$ на множестве X.

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполнено предположение 1 и, кроме того: (i) — множество Y ограничено и замкнуто; (ii) — множество X выпукло и замкнуто; (iii) — последовательность сеток* $\{Y_{N_k}\}_{k=0}^{\infty}$ *всюду плотна на множестве Y, т. е. для произвольного* $\varepsilon > 0$ *можно указать такой индекс* k_ε, *что для всех* $k > k_\varepsilon$ *расстояние между произвольной точкой* $y \in Y$ *и ближайшей к y точкой сетки* Y_{N_k} *будет меньше ε. Тогда любая предельная точка последовательности* $\{x^k\}_{k=0}^{\infty}$, *порожденной алгоритмом 1, является стационарной точкой функции* $\max\limits_{y \in Y} \varphi(x, y)$ *на множестве X.*

Библиографические указания. При написании параграфа использованы работы [238, 239, 127].

6.5. Метод штрафа в задаче поиска максимина

Задача 1. Найти $x^* = \arg\max\limits_{x \in X} \min\limits_{y \in Y} \varphi(x, y)$ для заданной функции $\varphi: R^n \times R^m \to R^1$ и заданных компактных множеств X и Y.

Предположение 1. Функция φ непрерывна на произведении компактов X и Y.

В методе штрафа вычисление вектора x^* и величины максимина

$$u^* = \max_{x \in X} \min_{y \in Y} \varphi(x, y),$$

сводится к решению последовательности задач условной максимизации по (x, u) функций

$$g_\tau(x, u, \alpha_k) = u - \alpha_k \int_Y |\min\{0, \varphi(x, y) - u\}|^\tau d\mu(y), \quad \tau > 0,$$

где $\tau > 0$ — произвольный параметр; $\alpha_k > 0$, $\lim\limits_{k \to \infty} \alpha_k = \infty$; μ — мера на множестве Y.

Если $(x^*(\alpha_k), u^*(\alpha_k))$ — решение задачи максимизации g_τ на множестве $X \times U$ (здесь U — любой отрезок, содержащий точку u^*), то предельные точки последовательности $\{x^*(\alpha_k), u^*(\alpha_k)\}_{k=0}^{\infty}$ являются решением задачи 1.

При $\tau \geqslant 1$ функция g_τ всегда вогнута по u; если, кроме того, $\varphi(x, y)$ вогнута по x при каждом $y \in Y$, то функция $g_\tau(x, u, \alpha_k)$ вогнута по (x, u). Если $\varphi(x, y)$ дифференцируема по x, то при $\tau > 1$ функция $g_\tau(x, u, \alpha_k)$ дифференцируема по (x, u):

$$\frac{\partial}{\partial x} g_\tau(x, u, \alpha_k) = \tau\alpha_k \int_Y |\min\{0, \varphi(x, y) - u\}|^{\tau-1} \times \frac{\partial}{\partial x}\varphi(x, y)\, d\mu(y);$$

$$\frac{\partial}{\partial u} g_\tau(x, u, \alpha_k) = 1 - \tau\alpha_k \int_Y |\min\{0, \varphi(x, y) - u\}|^{\tau-1} d\mu(y).$$

Алгоритм 1

Начало. I. Выбрать последовательность $\{\alpha_i\}_{i=0}^{\infty}$ коэффициентов штрафа такую, что

$$\alpha_i \geqslant 0, \quad i = 0,\ 1,\ 2,\ \ldots; \quad \lim_{i\to\infty} \alpha_i = \infty.$$

II. Задать меру μ в пространстве, содержащем Y, такую, что любое непустое пересечение Y с любым открытым множеством имеет положительную меру.

III. Выбрать параметр $\tau > 0$.

IV. Задать отрезок U, содержащий внутри себя значение

$$u^* = \max_{x\in X} \min_{y\in Y} \varphi(x,\ y).$$

В качестве U можно, например, выбрать отрезок

$$[\min_{(x,y)\in X\times Y} \varphi(x,\ y) - 1,\quad \max_{(x,y)\in X\times Y} \varphi(x,\ y) + 1].$$

Если выбрано $\tau \geqslant 1$, то можно положить $U = (-\infty,\ \infty)$.

V. Положить $k = 0$

Основной цикл. VI. Определить функцию

$$g_\tau(x,\ u,\ \alpha_k) = u - \alpha_k \int_Y |\min\{0,\ \varphi(x,\ y) - u\}|^\tau d\mu(y).$$

VII. Вычислить точку $(x^*(\alpha_k),\ u^*(\alpha_k)) \in X \times U$, удовлетворяющую условию

$$g_\tau(x^*(\alpha_k),\ u^*(\alpha_k),\ \alpha_k) = \max_{(x,u)\in X\times U} g_\tau(x,\ u,\ \alpha_k).$$

Для нахождения точки $(x^*(\alpha_k),\ u^*(\alpha_k))$ решается задача максимизации по $(x,\ u)$ функции $g_\tau(x,\ u,\ \alpha_k)$ на множестве $X \times U$, причем решение этой задачи на $(k-1)$-й итерации принимается за начальное приближение задачи максимизации в k-й итерации.

VIII. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 1. *Пусть выполняется предположение 1. Тогда любая предельная точка $(x^*,\ u^*)$ последовательности $\{x^*(\alpha_k),\ u^*(\alpha_k)\}_{k=0}^{\infty}$, порожденной алгоритмом 1, является решением задачи 1, т. е.*

$$u^* = \max_{x\in X} \min_{y\in Y} \varphi(x,\ y) = \min_{y\in Y} \varphi(x^*,\ y),$$

причем при всех $k = 0,\ 1,\ \ldots,\ \tau > 0$ справедливы неравенства

$$\min_{y\in Y} \varphi(x^*(\alpha_k),\ y) \leqslant u^* \leqslant g_\tau(x^*(\alpha_k),\ u^*(\alpha_k),\ \alpha_k) \leqslant u^*(\alpha_k).$$

Теорема 1'. *Если Y — параллелепипед m-мерного пространства R^m, μ — мера Лебега, а функция $\varphi(x,\ y)$ удовлетворяет условию Липшица по y равномерно относительно x, т. е.*

$$|\varphi(x,\ y) - \varphi(x,\ \bar{y})| \leqslant \gamma \|y - \bar{y}\|,\ \forall y,\ \bar{y} \in Y,\quad x \in X,$$

то для достаточно больших α_k справедлива оценка

$$0 \leqslant u^*(\alpha_k) - u^* \leqslant \beta(m,\ \tau)\,[\gamma^m/\alpha_k]^{1/(\tau+m-1)}$$

где $\beta(m,\ \tau)$ — константа, не зависящая от α_k и γ.

Если $Y = \{y^1, \ldots, y^l\}$ — *конечное множество,*

$$g_\tau(x, u, \alpha_k) = u - \alpha_k \sum_{i=1}^{l} |\min\{0, \varphi(x, y^i) - u\}|^\tau,$$

то

$$(\alpha_k \tau l)^{-1/(\tau-1)} \leqslant u^*(\alpha_k) - u^* \leqslant (\alpha_k \tau)^{-1/(\tau-1)} \quad при \quad \tau > 1;$$

$$u^*(\alpha_k) = u^* \quad при \quad \tau = 1, \quad \alpha_k > 1.$$

Замечание 1. Если задана точность ε определения величины максимина, то в алгоритме 1 вычисления следует проводить до тех пор, пока не будет выполнено неравенство

$$u^*(\alpha_k) - \min_{y \in Y} \varphi(x^*(\alpha_k), y) \leqslant \varepsilon$$

(для этого необходимо оценить глобальный минимум функции $\varphi(x^*(\alpha_k), y)$ по $y \in Y$).

Замечание 1'. Для ускорения поиска максимина с заданной точностью, сокращения объема вычислений, преодоления трудностей, связанных с «овражным» характером параметризованной задачи, в [372] рекомендуется последовательность α_k, $k = 0, 1, \ldots$, выбирать по правилу

$$\alpha_k < \alpha_{k+1} \leqslant \alpha_k + \beta(m, \tau, \gamma)\, \alpha_k^{(m+\tau-2)/\tau-1}, \quad k = 0, 1, \ldots,$$

где $\beta(m, \tau, \gamma) > 0$ — константа, не зависящая от коэффициента штрафа α_k; γ — константа Липшица функции $\varphi(x, y)$ по y; α_0 следует брать не слишком большим, чтобы уменьшить время поиска $(x^*(\alpha_0), u^*(\alpha_0))$.

Библиографические указания. Параграф написан на основании работ [67, 368, 372]. Дополнительные сведения о методе штрафов для решения задач максимина можно найти в работах [369, 370, 66].

6.6. Методы стохастического квазиградиента в задаче поиска максимина

З а д а ч а 0. Найти $x^* = \arg\max_{x \in X} \min_{y \in Y} \varphi(x, y)$ для заданной функции $\varphi : \boldsymbol{R}^n \times \boldsymbol{R}^m \to \boldsymbol{R}^1$, компактного множества Y пространства $\boldsymbol{R}^m$ и множества

$$X = \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\}. \qquad (6.3)$$

Предположения 0. (*i*) — функция $\varphi(x, y)$ — ограничена и непрерывно дифференцируема по x на $\boldsymbol{R}^n \times Y$; (*ii*) — функции $f_i(x)$, $i = 1, \ldots, m$ — непрерывно дифференцируемы и ограничены снизу на $\boldsymbol{R}^n$.

Задача вычисления вектора x^* и величины максимина

$$u^* = \max_{x \in X} \min_{y \in Y} \varphi(x, y)$$

сводится к максимизации по $(x, u) \in X \times U$ функции

$$g_{\tau_1,\tau_2}(x, u, \alpha) = u - \alpha \int_Y |\min\{0, \varphi(x, y) - u\}|^{\tau_1} d\mu(y) -$$

$$- \sum_{i=1}^{m} p_i |\min\{0, f_i(x)\}|^{\tau_2},$$

где $\tau_1, \tau_2 > 0$; $p_i > 0$; $\sum_{i=1}^{m} p_i = 1$; α — коэффициент штрафа; μ — некоторая мера на множестве Y; U — любой отрезок, содержащий точку u^*.

Функцию $g_{\tau_1,\tau_2}(x, u, \alpha)$ можно рассматривать как математическое ожидание $E_{y,i} q_{\tau_1,\tau_2}(x, u, \alpha/y, i)$ функции

$$q_{\tau_1,\tau_2}(x, u, \alpha/y, i) = u - \alpha |\min\{0, \varphi(x, y) - u\}|^{\tau_1} -$$

$$- \alpha |\min\{0, f_i(x)\}|^{\tau_2}$$

от двух независимых случайных величин y, i, причем y распределена на компакте Y в соответствии с мерой μ, а i принимает значения из множества $\mathcal{I} = \{1, \ldots, m\}$ с вероятностями $p_1, p_2, \ldots, p_m$.

Для решения задачи максимизации применяется метод стохастического градиента. В алгоритмах с ростом итераций коэффициент штрафа α стремится к бесконечности и в отличие от алгоритма 1 (см. § 6.5) на каждой итерации нет необходимости решать задачи максимизации и вычислять многомерные интегралы.

1. Основной алгоритм

Алгоритм 1

Начало. I. Выбрать произвольное начальное приближение $(x^1, u^1) \in R^n \times R^1$.

II. Задать меру μ в пространстве, содержащем Y, такую, что любое непустое пересечение Y с любым открытым множеством имеет положительную меру.

III. Выбрать параметры $\tau_1 \geqslant 2, \tau_2 \geqslant 2$ и константу $\alpha_0 > 0$ (обычно выбирают: $\tau_1 = 2$, $\tau_2 = 2$, $\alpha_0 \in [1, 10^2]$).

IV. Задать числа p_i, $i = 1, \ldots, m$ ($p_i > 0$, $i = 1, \ldots, m$, $\sum_{i=1}^{m} p_i = 1$), которые, соответственно, характеризуют требуемую относительную точность выполнения ограничений (6.3) в задаче 0 (обычно $p_i = 1/m$, $i = 1, \ldots, m$).

V. Положить $k = 1$.

Основной цикл. VI. Найти независимую реализацию y^k случайной величины $\tilde{y}$, распределенной на компакте Y в соответствии с мерой μ.

VII. Найти независимую реализацию i_k случайной величины $\tilde{i}$, которая принимает значения из множества $\mathcal{J} = \{1, \ldots, m\}$ с вероятностями $p_1, p_2, \ldots, p_m$, соответственно.

VIII. Вычислить значение шагового множителя ρ_k, удовлетворяющее условиям теоремы 1.

IX. Вычислить приращение Δ_k коэффициента штрафа, удовлетворяющее условиям теоремы 1.

X. Положить $\alpha_k = \alpha_{k-1} + \Delta_k$.

XI. Вычислить стохастический градиент (ξ^k, η^k) функции $g_{\tau_1,\tau_2}(x, u, \alpha_k)$ в точке (x^k, u^k)

$$\xi^k = \alpha_k\tau_1 |\min\{0, \varphi(x^k, y^k) - u^k\}|^{\tau_1-1} \nabla_x \varphi(x^k, y^k) + \\ + \alpha_k\tau_2 |\min\{0, f_{i_k}(x^k)\}|^{\tau_2-1} \nabla_x f_{i_k}(x^k); \quad (6.4)$$

$$\eta^k = 1 - \alpha_k\tau_1 |\min\{0, \varphi(x^k, y^k) - u^k\}|^{\tau_1-1}. \quad (6.5)$$

XII. Вычислить следующее приближение (x^{k+1}, u^{k+1}) по формулам

$$x^{k+1} = x^k + \rho_k\xi^k;$$
$$u^{k+1} = u^k + \rho_k\eta^k.$$

XIII. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 1. *Пусть выполняются предположения 0 и (iii) — $\nabla_x\varphi(x, y)$ ограничен на $R^n \times Y$ и удовлетворяет условию Липшица по x на R^n равномерно относительно $y \in Y$; (iv) — $\nabla_x f_i(x)$, $i = 1, \ldots, m$, ограничены и удовлетворяют условию Липшица на R^n; (v) — числовые последовательности $\{\rho_k\}_{k=1}^{\infty}$, $\{\Delta_k\}_{k=1}^{\infty}$ в алгоритме 1 такие, что*

$$\rho_k > 0, \quad \sum_{k=1}^{\infty} \rho_k = \infty, \quad \lim_{k\to\infty} \alpha_k = \infty, \quad \lim_{k\to\infty} \Delta_k = 0, \quad \Delta_k > 0;$$

$$\lim_{k\to\infty} (\Delta_k/\rho_k) = 0, \quad \lim_{k\to\infty} (\rho_k\alpha_k^3) = 0 \quad (\rho_k = k^{-1/2}, \quad \Delta_k = k^{-1}).$$

1. Тогда для любого начального приближения (x^1, u^1) последовательность $\{(x^k, u^k)\}_{k=1}^{\infty}$, порождаемая алгоритмом 1, содержит подпоследовательность такую, что все ее конечные предельные точки (x', u') с вероятностью 1 удовлетворяют следующим условиям: существуют числа

$$\beta_s, \quad s = 1, \ldots, n+1, \quad \lambda_i \geqslant 0, \quad i = 1, \ldots, m,$$

и точки

$$y_s, \quad s = 1, \ldots, n+1, \quad y_s \in \{\arg\min_{y\in Y} \varphi(x', y)\},$$

для которых справедливы соотношения

$$u' = \min_{y\in Y} \varphi(x', y);$$

$$\sum_{s=1}^{n+1} \beta_s + \sum_{i=1}^{m} \lambda_i > 0; \tag{6.6}$$

$$\sum_{s=1}^{n+1} \beta_s \nabla_x \varphi(x', y_s) + \sum_{i=1}^{m} \lambda_i \nabla_x f_i(x') = 0.$$

2. Если, кроме того,

$$\mathrm{co}\{\nabla_x f_i(x), \quad i \in \mathcal{I}^-(x)\} \not\ni 0, \tag{6.7}$$

где

$$\mathcal{I}^-(x) = \{i \mid f_i(x) \leqslant 0, \quad i \in \mathcal{I}\},$$

то $x' \in X$, $\sum_{s=1}^{n+1} \beta_s > 0$ *и справедливы условия дополняющей нежесткости*

$$\lambda_i f_i(x') = 0, \quad i = 1, \ldots, m,$$

т. е. точка x' *является стационарной в задаче 0.*

Замечание 1. Условие (6.7) выполнено для $x \in \boldsymbol{R}^n$, если векторы $\nabla_x f_i(x)$, $i \in \mathcal{I}^-(x)$, линейно-независимы для любых x таких, что $\mathcal{I}^-(x) \neq \varnothing$. Данное условие выполняется также в том случае, когда функции $f_i(x)$ вогнуты и удовлетворяют условию Слейтера на пространстве $\boldsymbol{R}^n$.

2. Следящий алгоритм

Для практического нахождения стационарных точек в задаче 0 с помощью алгоритма 1 необходимо выделять специальную последовательность, все предельные точки которой являются стационарными. В приводимом ниже алгоритме нужная подпоследовательность выделяется при реализации итерационного процесса.

Алгоритм 2

Начало. I—IV. Шаги I—IV такие, как и в алгоритме 1.

V. Выбрать произвольное значение $\varepsilon > 0$.

VI. Выбрать произвольный вектор $v^1 \in \boldsymbol{R}^{m+1}$ (обычно в качестве v^1 выбирают приближенное значение градиента функции $g_{\tau_1,\tau_2}(x, u, \alpha_1)$ по (x, u) в точке (x^1, u^1)).

VII. Положить $k = 1$, $j = 1$.

Основной цикл. VIII. Если $\|v^k\| < \varepsilon$, то положить

$$k_j = k, \quad \varepsilon = \varepsilon/2, \quad j = j + 1$$

и перейти к шагу IX, если $\|v^k\| \geqslant \varepsilon$, то перейти к шагу IX.

IX. Найти независимую реализацию y^k случайной величины $\tilde{y}$, распределенной на компакте Y в соответствии с мерой μ.

X. Найти независимую реализацию i_k случайной величины i, которая принимает значения из множества $\mathcal{I} = \{1, \ldots, m\}$ с вероятностями $p_1, p_2, \ldots, p_m$, соответственно.

XI. Вычислить значения шаговых множителей ρ_k, ρ'_k и значение коэффициента штрафа α_k, удовлетворяющее условиям теоремы 2.

XII. Вычислить стохастический градиент $\zeta^k = (\xi^k, \eta^k)$ функции $g_{\tau_1,\tau_2}(x, u, \alpha_k)$ в точке (x^k, u^k) по (6.4), (6.5).

XIII. Вычислить следующее приближение

$$x^{k+1} = x^k + \rho_k \xi^k; \quad u^{k+1} = u^k + \rho_k \eta^k.$$

XIV. Вычислить вектор

$$v^{k+1} = v^k + \rho'_k (\zeta^k - v^k).$$

XV. Положить $k = k + 1$ и перейти к шагу VIII.

Теорема 2. *Пусть выполняются предположения 0, условия (iii), (iv) теоремы 1 и числовые последовательности* $\{\rho_k\}_{k=1}^{\infty}$, $\{\rho'_k\}_{k=1}^{\infty}$, $\{\Delta_k\}_{k=1}^{\infty}$ *такие, что*

$$\rho_k > 0, \quad \lim_{k\to\infty} \rho_k = 0, \quad \sum_{k=1}^{\infty} \rho_k = \infty;$$

$$\lim_{k\to\infty} \alpha_k = \infty; \ \alpha_{k+1} = \alpha_k + \Delta_k; \ \Delta_k > 0, \ \lim_{k\to\infty} \Delta_k = 0, \ \sum_{k=1}^{\infty} (\Delta_k)^2 < \infty;$$

$$\lim_{k\to\infty} \Delta_k \alpha_k = 0, \ \lim_{k\to\infty} \rho'_k = 0, \ \sum_{k=1}^{\infty} \rho'_k = \infty, \ \lim_{k\to\infty} \Delta_k/\rho_k = 0;$$

$$\lim_{k\to\infty} \rho_k (\alpha_k)^3 = 0, \quad \sum_{k=1}^{\infty} (\rho'_k \alpha_k)^2 < \infty;$$

$$\sum_{k=1}^{\infty} (\rho'_k)^2 \alpha_k \left(\sum_{s=1}^{k} \rho'_s \alpha_s\right) < \infty, \quad \sum_{k=1}^{\infty} (\rho'_k)^2 \left(\sum_{s=1}^{k} \rho'_s \alpha_s\right)^2 < \infty;$$

$$\sum_{k=1}^{\infty} (\rho_k^2 \alpha_k^4/\rho'_k) < \infty, \quad \rho_k \alpha_k^4/\rho'_k > \rho_{k+1} \alpha_{k+1}^4/\rho'_{k+1}.$$

Тогда для любых векторов (x^1, u^1), v^1:

1) *последовательность* $\{(x^k, u^k)\}_{k=1}^{\infty}$, *порождаемая алгоритмом 2, содержит такую подпоследовательность* $\{(x^{s_i}, u^{s_i})\}_{i=1}^{\infty}$, *что почти наверное*

$$\lim_{i\to\infty} \nabla_{(x,u)} g_{\tau_1,\tau_2} (x^{s_i}, u^{s_i}, \alpha_{s_i}) = 0.$$

Любая предельная точка (x', u') *этой подпоследовательности удовлетворяет условиям (6.6) и является стационарной в задаче 0, если*

$$0 \notin \mathrm{co}\{\nabla_x f_i(x'), \quad i \in \mathcal{I}^-(x')\};$$

2) $\lim_{k\to\infty} \| v^k - \nabla_{(x,u)} g_{\tau_1,\tau_2}(x^k, u^k, \alpha_k) \| = 0$ п. н;

3) *предельная точка* $(\bar{x}, \bar{u})$ *последовательности* $\{(x^{k_j}, u^{k_j})\}_{j=1}^{\infty}$,

порожденной алгоритмом 2, является стационарной точкой в задаче 0.

Приведенные выше условия будут выполняться, если, например, положить

$$\rho_k = 1/k; \quad \rho_k' = k^{-12/13}; \quad \Delta_k = (k \ln k)^{-1},$$

или

$$\rho_k = k^{-49/50}; \quad \rho_k' = k^{-21/25}; \quad \Delta_k = k^{-99/100}.$$

3. Выпуклый случай

Задача 3. Найти $\arg\max_{x \in X} \min_{y \in Y} \varphi(x, y)$ для заданной функции $\varphi : R^n \times R^m \to R^1$, компактного множества $Y \subset R^m$ и множества

$$X \triangleq \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m; \quad x \in Z\},$$

где Z — выпуклый компакт в R^n; $f_i : R^n \to R^1$, $i = 1, \ldots, m$ — вогнутые функции на Z.

Алгоритм 3

Начало. I—V. Шаги I—V такие, как в алгоритме 1.

Основной цикл. VI. Найти независимую реализацию y^k случайной величины $\tilde{y}$, распределенной на компакте Y в соответствии с мерой μ.

VII. Найти независимую реализацию i_k случайной величины $\tilde{i}$, которая принимает значения из множества $\mathcal{I} = \{1, \ldots, m\}$ с вероятностями $p_1, p_2, \ldots, p_m$, соответственно.

VIII. Найти значения шагового множителя ρ_k и коэффициента штрафа α_k, удовлетворяющие условиям теоремы 3.

IX. Вычислить стохастический градиент (ξ^k, η^k) функции $g_{\tau_1,\tau_2}(x, u, \alpha_k)$ в точке (x^k, u^k) по формулам (6.4), (6.5).

X. Вычислить следующее приближение (x^{k+1}, u^{k+1})

$$x^{k+1} = \pi_Z(x^k + \rho_k \xi^k); \quad u^{k+1} = \pi_U(u^k + \rho_k \eta^k),$$

где π_Q — оператор проектирования на Q; U — достаточно большой отрезок, содержащий

$$[\min_{(x,y) \in Z \times Y} \varphi(x, y), \max_{(x,y) \in Z \times Y} \varphi(x, y)].$$

XI. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 3. *Пусть выполняются условия: (**i**) — Z — выпуклый компакт, Y — компакт; (**ii**) — функции $\varphi(x, y)$, $\nabla_x \varphi(x, y)$ непрерывны на $Z \times Y$; (**iii**) — $\varphi(x, y)$ вогнута по x на Z для любого $y \in Y$; (**iv**) — функции $f_i(x)$, $i = 1, \ldots, m$, имеют непрерывные частные производные и вогнуты на Z; (**v**) — числовые последователь-*

ности $\{\rho_k\}_{k=1}^{\infty}$, $\{\alpha_k\}_{k=1}^{\infty}$ *такие, что*

$$\rho_k > 0,\ \lim_{k\to\infty}\rho_k = 0,\ \rho_{k+1} \leqslant \rho_k,\ \sum_{k=1}^{\infty}\rho_k = \infty;$$

$$\alpha_k > 0,\ \lim_{k\to\infty}\alpha_k = \infty,\ \alpha_{k+1} \geqslant \alpha_k,\ \sum_{k=1}^{\infty}(\rho_k\alpha_k)^2 < \infty$$

(условия (v) будут выполняться, если положить

$$\rho_k = k^{-3/4},\quad \alpha_k = k^{1/6},\quad k = 1,\ 2,\ \ldots).$$

Тогда для любого начального приближения $(x^1,\ u^1)$ *последовательность* $\{(x^k,\ u^k)\}_{k=1}^{\infty}$, *порожденная алгоритмом 3, с вероятностью 1 сходится к множеству* G^* *решений задачи 3:*

$$G^* \triangleq \{\arg\max_{x\in X}\min_{y\in Y}\varphi(x,\ y)\} \times \{u^*\}.$$

Замечание 3. Если в максиминной задаче 3 $Z \equiv \boldsymbol{R}^n$ и функции $\varphi(x,\ y)$, $f_i(x)$, $i = 1, \ldots, m$, удовлетворяют условиям теоремы 1, то на шаге X алгоритма 3 точку $(x^{k+1},\ u^{k+1})$ можно вычислять по формулам

$$x^{k+1} = x^k + \rho_k\xi^k;\quad u^{k+1} = u^k + \rho_k\eta^k,$$

где ρ_k, α_k вычисляют в соответствии с требованиями теоремы 3.

Замечание 3'. Алгоритмы, приведенные в данном параграфе, сравнительно просты для реализации, устойчивы относительно ошибок вычислений; их эффективность по сравнению с другими методами решения задачи 0 увеличивается с ростом размерностей переменных x и y. Решение тестовых задач на ЭВМ свидетельствует о более быстрой сходимости алгоритмов по переменной x, чем по переменной u, поэтому в [372] рекомендуется один шаг по переменной x чередовать с несколькими шагами по переменной u.

Библиографические указания. Пункт 1 написан на основании работ [372, 165], пункт 2 — на основании работ [372, 164]. При написании пункта 3 использовались работы [372, 165, 266].

6.7. Метод невязок в задаче поиска максимина

Задача 1. Найти $x^* = \arg\max_{x\in X}\min_{y\in Y}\varphi(x,\ y)$ для заданной функции $\varphi : \boldsymbol{R}^n \times \boldsymbol{R}^m \to \boldsymbol{R}^1$, заданного параллелепипеда $Y \subset \boldsymbol{R}^m$ и множества

$$X \triangleq \{x \mid f_i(x) \geqslant 0,\quad \psi_j(x) = 0,\quad i = 1,\ \ldots,\ l_1,\quad j = 1,\ \ldots,\ l_2,\quad x \in Q\},$$

где Q — заданный компакт.

Предположения 1. Функции $f_i(x)$, $i = 1, \ldots, l_1$, $\psi_j(x)$, $j = 1, \ldots, l_2$ — непрерывны по x; $\varphi(x,\ y)$ — непрерывна по $(x,\ y)$.

В методе «невязок» вводится функция

$$g(x, u) = \int_Y [\min\{0, \varphi(x, y) - u\}]^2 dy +$$

$$+ \sum_{j=1}^{l_2} \beta_j (\psi_j(x))^2 + \sum_{i=1}^{l_1} \alpha_i [\min\{0, f_i(x)\}]^2,$$

где $\beta_j > 0, \alpha_i > 0$ — произвольные константы.

Значение максимина

$$u^* \triangleq \max_{x \in X} \min_{y \in Y} \varphi(x, y)$$

вычисляется как максимальное u, для которого

$$\min_{x \in Q} g(x, u) = 0,$$

при этом $\bar{x} = \arg\min_{x \in Q} g(x, u^*)$ является решением задачи 1. На каждой итерации алгоритма требуется решать задачу минимизации по $x \in Q$ функции $g(x, u^k)$, где $\{u^k\}_{k=0}^{\infty}$ сходится к u^*, что представляет собой сложную задачу из-за трудностей, связанных с вычислением многомерных интегралов.

В работе [72] для вычисления многомерных интегралов использован один из вариантов метода Монте-Карло. Метод «невязок» для решения максиминных задач следует использовать, когда x и y имеют небольшую размерность, или когда Y состоит из конечного числа точек (тогда вычисление интеграла заменяется вычислением суммы).

Алгоритм 1

Начало. I. Выбрать произвольное значение $u^0 \in R^1$ (обычно u^0 выбирают из отрезка $[\min_{(x,y) \in Q \times Y} \varphi(x, y), \max_{(x,y) \in Q \times Y} \varphi(x, y)]$) и произвольное начальное приближение $\bar{x}^0 \in Q$.

II. Выбрать приращение $\Delta > 0$.

III. Выбрать произвольные константы $\alpha_i > 0, \ i = 1, \ldots, l_1$; $\beta_j > 0, \ j = 1, \ldots, l_2$.

IV. Определить функцию

$$g(x, u) = \int_Y [\min\{0, \varphi(x, y) - u\}]^2 dy +$$

$$+ \sum_{j=1}^{l_2} \beta_j (\psi_j(x))^2 + \sum_{i=1}^{l_1} \alpha_i [\min\{0, f_i(x)\}]^2. \quad (6.8)$$

V. Положить $k = 0$.

VI. Вычислить

$$\sigma_0 = \min_{x \in Q} g(x, u^0)$$

и вектор

$$x^0 = \arg\min_{x \in Q} g(x, u^0), \quad (x^0 \in Q).$$

Если $\sigma_0 = 0$, то перейти к шагу VII, если $\sigma_0 > 0$, то — к шагу XI.

Основной цикл. VII. Положить $\tau = 1$.

VIII. Вычислить

$$u^{k+1} = u^k + \tau\Delta.$$

IX. Вычислить

$$\sigma_{k+1} = \min_{x \in Q} g(x, u^{k+1}) \tag{6.9}$$

и вектор $x^{k+1} \in Q$

$$x^{k+1} = \arg\min_{x \in Q} g(x, u^{k+1}). \tag{6.10}$$

X. Если $\sigma_{k+1} = 0$, то положить $k = k + 1$, $\tau = \tau + 1$ и перейти к шагу VIII; иначе (если $\sigma_{k+1} > 0$) положить $\eta_{k+1}^{-} = u^k$, $\eta_{k+1}^{+} = u^{k+1}$, $k = k + 1$ и перейти к шагу XV.

XI. Положить $\tau = 1$.

XII. Вычислить

$$u^{k+1} = u^k - \tau\Delta.$$

XIII. Вычислить σ_{k+1} и x^{k+1}, соответственно, по (6.9), (6.10).

XIV. Если $\sigma_{k+1} > 0$, то положить $k = k + 1$, $\tau = \tau + 1$ и перейти к шагу XII; иначе (если $\sigma_{k+1} = 0$) положить

$$\eta_{k+1}^{-} = u^{k+1}, \quad \eta_{k+1}^{+} = u^k, \quad k = k + 1$$

и перейти к шагу XV.

XV. Вычислить

$$u^{k+1} = (\eta_k^{-} + \eta_k^{+})/2.$$

XVI. Вычислить σ_{k+1}, x^{k+1}, соответственно по (6.9), (6.10).

XVII. Если $\sigma_{k+1} = 0$, то положить $\eta_{k+1}^{-} = u^{k+1}$, $\eta_{k+1}^{+} = \eta_k^{+}$, $k = k + 1$ и перейти к шагу XV; иначе (т. е. если $\sigma_{k+1} > 0$) положить $\eta_{k+1}^{-} = \eta_k^{-}$, $\eta_{k+1}^{+} = u^{k+1}$, $k = k + 1$ и перейти к шагу XV.

Теорема 1. *Если выполнено предположение 1, то алгоритм 1 порождает последовательность* $\{u^k\}_{k=0}^{\infty}$, *для которой*

$$\lim_{k \to \infty} u^k = u^*,$$

причем точка

$$\bar{x} = \arg\min_{x \in Q} g(x, u^*)$$

является решением задачи 1, т. е.

$$\bar{x} = \arg\max_{x \in X} \min_{y \in Y} \varphi(x, y).$$

Замечание 1. На практике итерации проводят до тех пор, пока для заданного малого числа $\varepsilon > 0$ не выполнится неравенство

$$\eta_k^{+} - \eta_k^{-} < \varepsilon.$$

Тогда за приближенное значение максимина принимают u^k, а за решение задачи 1 — вектор $x^k \in Q$,

$$x^k = \arg\min_{x \in Q} g(x, u^k).$$

Библиографические указания. При написании параграфа использовались работы [70, 72, 372].

6.8. Квазиградиентные методы решения непрерывных минимаксных задач стохастического программирования

Задача 0. Найти $\arg\min_{x \in X} \max_{y \in Y} E\varphi(x, y, \omega)$ для заданной функции $\varphi : R^n \times R^m \times \Omega \to R^1$ и заданных множеств $X \subset R^n$, $Y \subset R^m$.

Предположения 0. (*i*) — функция $f(x, y) \triangleq E\varphi(x, y, \omega)$ и ее градиент по x непрерывны по совокупности переменных x, y; (*ii*) — X — выпуклое, замкнутое ограниченное множество; (*iii*) — Y — замкнутое ограниченное множество; (*iv*) — имеется возможность вычислять только значения функции $\varphi(x, y, \omega)$ и ее градиента $\nabla_x\varphi(x, y, \omega)$ в каждой точке $(x, y, \omega) \in X \times Y \times \Omega$.

Предлагаемые ниже методы решения задачи 0 основаны на построении специальных множеств Y_k для определения на k-й итерации направления движения h^k к $(k+1)$-му приближению x^{k+1}. Вектор h^k определяется как стохастический квазиградиент функции $\max_{y \in Y_k} E\varphi(x, y, \omega)$ в точке x^k.

1. Стохастический квазиградиентный метод

Алгоритм 1

Начало. I. Выбрать начальное приближение $x^0 \in X$.

II. Выбрать натуральное число N_s, положить $N = N_s$

III. Выбрать произвольные действительные числа z_i^0, $i \in [0 : N]$.

IV. Построить сетку

$$Y_N = \{y^i \mid y^i \in Y, \quad i \in [0 : N]\}.$$

V. Положить $k = 0$.

Основной цикл. VI. Найти индекс i_k из условия

$$z_{i_k}^k = \max_{i \in [0:N]} z_i^k.$$

VII. Вычислить независимую реализацию ω^k случайного параметра ω.

VIII. Вычислить $\nabla_x\varphi(x^k(\omega), y^{i_k}, \omega^k)$.

IX. Вычислить шаговые множители ρ_k и σ_k, удовлетворяющие условиям теоремы 1.

X. Вычислить следующее приближение

$$x^{k+1}(\omega) = \pi_X(x^k(\omega) - \rho_k\nabla_x\varphi(x^k(\omega), y^{i_k}, \omega^k)).$$

XI. Вычислить значения функции $\varphi(x^{k+1}(\omega),\ y^i,\ \omega^k)$, $i \in [0:N]$.

XII. Вычислить величины

$$z_i^{k+1}(\omega) = z_i^k(\omega) + \sigma_k(\varphi(x^{k+1}(\omega),\ y^i,\ \omega^k) - z_i^k(\omega)), \quad i \in [0:N].$$

XIII. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 1. *Если выполнены предположения 0 и (i) — функция $E\varphi(x, y, \omega)$ выпукла по x, $x \in X$ при каждом $y \in Y$; (**ii**) — градиент функции $f(x, y) \triangleq E\varphi(x, y, \omega)$ по x удовлетворяет условию Липшица*

$$\|\nabla_x f(x, y) - \nabla_x f(\bar{x}, y)\| \leqslant \alpha_1 \|x - \bar{x}\|, \quad y \in Y, \quad x, \bar{x} \in X;$$

(***iii***) —

$$E|\varphi(x, y, \omega)|^2 < \infty, \quad E\|\nabla_x \varphi(x, y, \omega)\|^2 < \infty, \quad y \in Y, \quad x \in X;$$

(***iv***) — *шаговые множители ρ_k и σ_k удовлетворяют условиям*

$$\sum_{k=0}^{\infty} \sigma_k^2 < \infty, \quad \sum_{k=0}^{\infty} \rho_k = \infty, \quad \rho_k/\sigma_k \to 0 \ \ \text{при} \ \ k \to \infty;$$

$$\rho_{k+1}/\rho_k \to 1 \ \ \text{при} \ \ k \to \infty,$$

то почти при каждом ω предельные точки $\bar{x}^{N_s}$ последовательности $\{x^k(\omega)\}_{k=0}^{\infty}$, порожденной алгоритмом 1, являются точками минимума функции $\max\limits_{i \in [0:N]} E\varphi(x, y^i, \omega)$ на множестве X.

Для решения исходной задачи 0, необходимо при каждом фиксированном N_s ($N_s \to \infty$ при $s \to \infty$) решать дискретную мичимаксную задачу с помощью алгоритма 1, причем последовательность сеток $\{Y_{N_s}\}_{s=0}^{\infty}$ должна обладать свойством плотности на множестве Y. Каждая предельная точка последовательности $\{\bar{x}^{N_s}\}_{s=0}^{\infty}$ является решением задачи 0.

Последовательность сеток $\{Y_{N_s}\}_{s=0}^{\infty}$ «обладает свойством плотности» на множестве Y, если для любого $\varepsilon > 0$ можно указать такой номер $\bar{N}$, что для всех $N_s > \bar{N}$ расстояние между произвольной точкой $\bar{y} \in Y$ и ближайшей к $\bar{y}$ точкой сетки Y_{N_s} будет меньше ε.

2. Модифицированный стохастический квазиградиентный метод

Алгоритм 2

Начало. I. Выбрать начальное приближение $x^0 \in \boldsymbol{R}^n$.

II. Выбрать натуральное число N_0 и построить сетку

$$Y_{N_0} = \{y^i \mid y^i \in Y, \quad i \in [0:N_0]\}.$$

III. Выбрать произвольные действительные числа z_i^0, $i \in [0:N_0]$.

IV. Выбрать начальное значение шагового множителя $\rho_0 > 0$ и константы $\alpha > 0$, $\beta > 0$.

V. Положить $k = 0$.

Основной цикл. VI. Найти индекс i_k из условия

$$z_{i_k}^k(\omega) = \max_{i \in [0:N_k]} z_i^k(\omega).$$

VII. Найти независимую реализацию ω^k случайного параметра ω.

VIII. Вычислить $\nabla_x (\varphi (x^k(\omega), y^{i_k}, \omega^k)$.

IX. Вычислить следующее приближение

$$x^{k+1}(\omega) = \pi_X (x^k(\omega) - \rho_k \nabla_x \varphi (x^k(\omega), y^{i_k}, \omega^k)).$$

X. Вычислить значения $\varphi (x^{k+1}(\omega), y^i, \omega^k)$, $i \in [0 : N_k]$.

XI. Найти шаговые множители ρ_{k+1} и σ_{k+1}, удовлетворяющие условиям теоремы 2.

XII. Построить сетку

$$Y_{N_{k+1}} = \{y^i \mid y^i \in Y, \quad i \in [0 : N_{k+1}]\}$$

так, чтобы она содержала все точки сетки Y_{N_k} с сохранением их нумерации и чтобы выполнялось следующее условие: для произвольного $\bar{y} \in Y$ расстояние от $\bar{y}$ до ближайшей точки сетки $Y_{N_{k+1}}$ не превышало значения $\alpha\rho_{k+1}^{1+\beta}$.

XIII. Вычислить величины

$$z_i^{k+1}(\omega) = z_i^k(\omega) + \sigma_{k+1}(\varphi(x^{k+1}(\omega), y^i, \omega^k) - z_i^k(\omega)) \quad i \in [0 : N_k].$$

XIV. Положить $i = N_k + 1$.

XV. Найти точку y^{j_i} сетки Y_{N_k}, которая принадлежит шару радиусом $\alpha\rho_{k+1}$ с центром в точке y^i и имеет наименьший индекс $j_i \in [0 : N_k]$.

XVI. Положить $z_i^{k+1}(\omega) = z_{j_i}^k(\omega)$.

XVII. Если $i < N_{k+1}$, то положить $i = i + 1$ и перейти к шагу XV; иначе перейти к шагу XVIII.

XVIII. Положить $k = k + 1$ и перейти к шагу VI.

Теорема 2. *Если выполнены условия теоремы 1 и (i) — функция $E\varphi(x, y, \omega)$ для произвольного $x \in X$ удовлетворяет условию Липшица по переменной y*

$$|E\varphi(x, y, \omega) - E\varphi(x, \bar{y}, \omega)| \leqslant \alpha_1 \|y - \bar{y}\|, \quad \alpha_1 < \infty;$$

(ii) — последовательность $\{\sigma_k\}_{k=0}^{\infty}$ дополнительно удовлетворяет условию

$$\lim_{s \to \infty} \sum_{j=s}^{m} \sigma_j = \infty \quad \textit{при} \quad m - s \to \infty,$$

(например $\sigma_k = k^{-\nu}$, $\nu \in (^1/_2, 1)$),

то почти при каждом ω предельные точки последовательности

$\{x^k(\omega)\}_{k=0}^{\infty}$, *порожденной алгоритмом 2, являются точками минимума функции* $\max_{y \in Y} \boldsymbol{E}\varphi(x, y, \omega)$ *на множестве* X.

Библиографические указания. Параграф написан на основании работ [30, 31, 32, 171, 113].

6.9. Градиентные методы отыскания седловых точек

Задача 0. Найти точку $(x^*, y^*) \in X \times Y$, удовлетворяющую условию

$$\varphi(x, y^*) \leqslant \varphi(x^*, y^*) \leqslant \varphi(x^*, y), \quad \forall x \in X, \quad y \in Y, \tag{6.11}$$

где $\varphi(x, y)$ — вогнуто-выпуклая функция; X и Y — выпуклые замкнутые телесные подмножества эвклидовых пространств $\boldsymbol{R}^n$ и $\boldsymbol{R}^m$, соответственно.

Определение 1. Множество точек (x^*, y^*), удовлетворяющих условию (6.11), называют множеством седловых точек $X^* \times Y^*$ функции $\varphi(x, y)$ на множестве $X \times Y$.

Предположения 0. (i) — функция $\varphi(x, y)$ непрерывно дифференцируема на множестве $X \times Y$, вогнута по x и выпукла по y; (ii) — множество седловых точек $X^* \times Y^*$ функции φ непусто и ограничено; (iii) — выполняется условие устойчивости множества седловых точек по x, т. е. для произвольного $x \in (X \setminus X^*)$ выполняется строгое неравенство

$$\varphi(x, y^*) < \varphi(x^*, y^*).$$

Предположения 0 выполняются для функции Лагранжа задачи вогнутого программирования:

найти
$$\arg\max_{x \in X_1} f_0(x), \tag{6.12}$$

где

$$X_1 = \{x \mid f_i(x) \geqslant 0, \quad i = 1, \ldots, m, \quad x \in \boldsymbol{R}^n\},$$

с дифференцируемыми (и вогнутыми) функциями f_j, $j = 0, 1, .., m$, если функция f_0 строго вогнута и выполняется условие Слейтера.

Условие (iii) играет существенную роль, так как при его отсутствии градиентные методы отыскания седловых точек, в общем, не сходятся. Примером служит функция

$$\varphi(x, y) = (1 - y)x, \quad X = \boldsymbol{R}^1, \quad Y = \boldsymbol{R}^1_+,$$

являющаяся функцией Лагранжа задачи

найти
$$\arg\max_{x \in X_2} x,$$

где

$$X_2 = \{x \mid -x \geqslant 0, \quad x \in \boldsymbol{R}^1\};$$

$\boldsymbol{R}^m_+$ — подпространство m-мерных векторов с неотрицательными компонентами.

1. Основной алгоритм

Алгоритм 1

Начало. I. Выбрать начальное приближение $(x^0, y^0) \in X \times \times Y$.

II. Положить $k = 0$

Основной цикл. III. Выбрать шаговый множитель ρ_k, удовлетворяющий условиям теоремы 1.

IV. Вычислить векторы $\nabla_x \varphi(x^k, y^k)$ и $\nabla_y \varphi(x^k, y^k)$ — градиенты, соответственно, по x и y функции $\varphi(x, y)$ в точке (x^k, y^k).

V. Вычислить следующие приближения:

$$x^{k+1} = \pi_X(x^k + \rho_k \nabla_x \varphi(x^k, y^k));$$
$$y^{k+1} = \pi_Y(y^k - \rho_k \nabla_y \varphi(x^k, y^k)),$$

где π_Q — оператор проектирования на множество Q.

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 1. *Пусть выполняются предположения 0. Тогда для любого положительного числа* δ *существует такое положительное число* $\rho(\delta)$, *что при всяком начальном приближении* $(x^0, y^0) \in Z(\delta)$,

$$Z(\delta) \triangleq \{(x, y) \mid \min_{(x^*, y^*) \in X^* \times Y^*} \|(x, y) - (x^*, y^*)\| \leqslant \delta, \quad (x, y) \in X \times Y\}, \quad (6.13)$$

и любой последовательности $\{\rho_k\}_{k=0}^{\infty}$, *удовлетворяющей условиям*

$$\rho_k > 0; \quad \rho_k \to 0; \quad \sum_{k=0}^{\infty} \rho_k = \infty; \quad \rho_k \leqslant \rho(\delta),$$

алгоритм 1 порождает последовательность $\{(x^k, y^k)\}_{k=0}^{\infty}$, *которая сходится к множеству седловых точек* $X^* \times Y^*$.

2. Градиентный метод отыскания седловых точек с постоянным шаговым множителем

Алгоритм 2

Начало. I. Выбрать произвольное начальное приближение $(x^0, y^0) \in X \times Y$.

II. Выбрать постоянный шаговый множитель $\rho > 0$.

III. Положить $k = 0$.

Основной цикл. IV. Вычислить векторы $\nabla_x \varphi(x^k, y^k)$ и $\nabla_y \varphi(x^k, y^k)$.

V. Вычислить следующие приближения:

$$x^{k+1} = \pi_X(x^k + \rho \nabla_x \varphi(x^k, y^k));$$
$$y^{k+1} = \pi_Y(y^k - \rho \nabla_y \varphi(x^k, y^k)),$$

где π_Q — оператор проектирования на множество Q.

VI. Положить $k = k + 1$ и перейти к шагу IV.

Теорема 2. *Пусть выполняются предположения 0. Тогда для любых чисел* ε *и* δ, *удовлетворяющих неравенству* $0 < \varepsilon < \delta$, *существуют такие положительные числа* $\rho(\varepsilon, \delta)$ *и* $k_0(\varepsilon, \delta)$, *что*

алгоритм 2 при всяком $(x^0, y^0) \in Z(\delta)$ *и произвольном постоянном шаговом множителе* $\rho \leqslant \rho(\varepsilon, \delta)$ *порождает последовательность* $\{(x^k, y^k)\}_{k=0}^{\infty}$, *для которой при всех* $k > k_0(\varepsilon, \delta)\rho^{-1}$ *будет* $(x^k, y^k) \in Z(\varepsilon)$ (здесь и далее множества $Z(\delta)$ и $Z(\varepsilon)$ определяются по (6.13)).

Теорема 2'. *Пусть выполняются предположения 0 и пусть:* (**iv**) — *множество седловых точек* $X^* \times Y^*$ *функции* $\varphi(x, y)$ *состоит из одной точки* (x^*, y^*); (**v**) — *функция* $\varphi(x, y)$ *имеет в точке* (x^*, y^*) *вторую производную;* (**vi**) — *точка* (x^*, y^*) *является внутренней точкой множества* $X \times Y$; (**vii**) — *не существует числа* λ *и ненулевого вектора* $(u, v) \in \boldsymbol{R}^n \times \boldsymbol{R}^m$ *таких, что*

$$\nabla^2_{xx}\varphi(x^*, y^*)u = 0, \quad \nabla^2_{yy}\varphi(x^*, y^*)v = 0;$$

$$\nabla^2_{yx}\varphi(x^*, y^*)u = \lambda v, \quad (\nabla^2_{yx}\varphi(x^*, y^*))^T v = \lambda u.$$

Тогда для любого числа $\delta > 0$ *найдется такое число* $\rho(\delta) > 0$, *что алгоритм 2 при всяком начальном приближении* $(x^0, y^0) \in Z(\delta)$ *и произвольном постоянном шаговом множителе* $\rho \leqslant \rho(\delta)$ *порождает последовательность* $\{(x^k, y^k)\}_{k=0}^{\infty}$, *которая линейно (со скоростью геометрической прогрессии) сходится к точке* (x^*, y^*).

Теорема 2''. *Пусть выполняются предположения теоремы 2', кроме условия* (**vi**), *и пусть:* (**ix**) — $X = \boldsymbol{R}^n$, $Y = \boldsymbol{R}^m_+$; (**x**) —

$$(\nabla_y\varphi(x^*, y^*), e^1) = 0, \ \ldots, \ (\nabla_y\varphi(x^*, y^*), e^p) = 0;$$

$$(\nabla_y\varphi(x^*, y^*), e^{p+1}) > 0, \ \ldots, \ (\nabla_y\varphi(x^*, y^*), e^m) > 0,$$

где e^i, $i = 1, \ldots, m$ — i-й *орт в пространстве* $\boldsymbol{R}^m$;
(**xi**) — *не существует вектора* $u \neq 0$ *и такого, что*

$$\nabla^2_{xx}\tilde{\varphi}(x^*, \tilde{y}^*)u = 0,$$

и не существует вектора $v \neq 0$ *и такого, что*

$$(\nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*))^T v = 0, \quad \nabla^2_{\tilde{y}\tilde{y}}\tilde{\varphi}(x^*, \tilde{y}^*)v = 0,$$

(*здесь* $\tilde{y}$ *обозначает проекцию точки* y *на подпространство* $\boldsymbol{R}^p$, *отвечающее первым* p *координатам;* $\tilde{\varphi}$ — *ограничение функции* φ *на множество* $\boldsymbol{R}^n \times \boldsymbol{R}^p_+$; $(\nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*))^T$ — *матрица, транспонированная к матрице* $\nabla^2_{\tilde{y}x}\tilde{\varphi}(x^*, \tilde{y}^*)$).

Тогда для любого $\delta > 0$ *найдется число* $\rho(\delta) > 0$ *такое, что алгоритм 2 при всяком начальном приближении* $(x^0, y^0) \in Z(\delta)$ *и произвольном постоянном шаговом множителе* $\rho \leqslant \rho(\delta)$ *порождает последовательность* $\{(x^k, y^k)\}_{k=0}^{\infty}$, *которая линейно (со скоростью геометрической прогрессии) сходится к седловой точке* (x^*, y^*).

Замечание 2. Если $\varphi(x, y)$ — функция Лагранжа задачи вогну-

того программирования (6.12), т. е.

$$\varphi(x, y) = f_0(x) + \sum_{i=1}^{m} y_i f_i(x),$$

то условия (*xi*) теоремы 2″ означают линейную независимость градиентов $\nabla f_1(x^*)$, ..., $\nabla f_p(x^*)$ активных ограничений и сильную выпуклость функции $-\varphi(x, y^*)$ в окрестности точки x^*.

3. Обобщенный градиентный метод

Задача 3. Найти точку $(x^*, y^*) \in X \times Y$, удовлетворяющую условию

$$\varphi(x, y^*) \leqslant \varphi(x^*, y^*) \leqslant \varphi(x^*, y), \quad \forall x \in X, \quad y \in Y,$$

где $\varphi(x, y)$ — вогнуто-выпуклая функция; X и Y — выпуклые и замкнутые множества эвклидовых пространств R^n и R^m, соответственно.

Предположения 3. (*i*) — функция $\varphi(x, y)$ при любом фиксированном $y \in Y$ выпукла вверх по x в некоторой окрестности $\tilde{X}$ множества X; функция $\varphi(x, y)$ при любом фиксированном $x \in X$ выпукла вниз по y в некоторой окрестности $\tilde{Y}$ множества Y; (*ii*) — множество седловых точек $X^* \times Y^*$ функции $\varphi(x, y)$ относительно множеств X и Y непусто и ограничено; (*iii*) — выполняется условие устойчивости множества седловых точек $X^* \times Y^*$ функции $\varphi(x, y)$, т. е. для произвольных $x^* \in X^*$, $y^* \in Y^*$ выполняется

$$X^* = \{x \mid \varphi(x, y^*) = \max_{x' \in X} \varphi(x', y^*), \quad x \in X\};$$

$$Y^* = \{y \mid \varphi(x^*, y) = \min_{y' \in Y} \varphi(x^*, y'), \quad y \in Y\}.$$

В приведенном ниже алгоритме в качестве векторов, определяющих направление движения по переменным x и y на k-й итерации, выбирают, соответственно, обобщенные градиенты функции $\varphi(x, y)$ по x и y, вычисленные в точке (x^k, y^k). Шаговые множители удовлетворяют классическим условиям.

Алгоритм 3

Начало. I. Выбрать произвольное начальное приближение $(x^0, y^0) \in X \times Y$.

II. Положить $k = 0$.

Основной цикл. III. Вычислить обобщенные градиенты $\hat{\nabla}_x \varphi(x^k, y^k)$ и $\hat{\nabla}_y \varphi(x^k, y^k)$ функции $\varphi(x, y)$ по x и y в точке (x^k, y^k), т. е. такие векторы, что

$$(\hat{\nabla}_x \varphi(x^k, y^k), x' - x^k) \geqslant \varphi(x', y^k) - \varphi(x^k, y^k), \quad \forall x' \in \tilde{X};$$

$$(\hat{\nabla}_y \varphi(x^k, y^k), y' - y^k) \leqslant \varphi(x^k, y') - \varphi(x^k, y^k), \quad \forall y' \in \tilde{Y}.$$

IV. Вычислить шаговый множитель ρ_k.

V. Вычислить следующее приближение (x^{k+1}, y^{k+1})

$$x^{k+1} = \pi_X(x^k + \rho_k \hat{\nabla}_x \varphi(x^k, y^k));$$

$$y^{k+1} = \pi_Y(y^k - \rho_k \hat{\nabla}_y \varphi(x^k, y^k)).$$

VI. Положить $k = k + 1$ и перейти к шагу III.

Теорема 3. *Пусть выполняются предположения 3 и пусть:* (**iv**) — *множества обобщенных градиентов функции $\varphi(x, y)$ по x и y, соответственно, равномерно ограничены, т. е. существует постоянная α такая, что*

$$\|\hat{\nabla}_x \varphi(x, y)\| \leqslant \alpha, \quad \|\hat{\nabla}_y \varphi(x, y)\| \leqslant \alpha, \quad \forall (x, y) \in X \times Y;$$

(**v**) — *шаговые множители ρ_k в алгоритме 3 удовлетворяют условиям*

$$\rho_k > 0, \quad k = 0, 1, \ldots; \quad \lim_{k\to\infty} \rho_k = 0; \quad \lim_{k\to\infty} \sum_{i=0}^{k} \rho_i = \infty.$$

Тогда последовательность $\{(x^k, y^k)\}_{k=0}^{\infty}$, порожденная алгоритмом 3, такова, что

$$\min_{(x^*, y^*) \in X^* \times Y^*} \|(x^k, y^k) - (x^*, y^*)\| \to 0 \quad \text{при} \quad k \to \infty. \tag{6.14}$$

Теорема 3'. *Пусть выполняются все предположения теоремы 3 за исключением условия* (**iv**). *Тогда существует число $\rho > 0$ такое, что при выполнении дополнительного условия*

$$\rho_k < \rho, \quad k = 0, 1, \ldots,$$

последовательность $\{(x^k, y^k)\}_{k=0}^{\infty}$, порожденная алгоритмом 3, удовлетворяет предельному соотношению (6.14).

Библиографические указания. Пункты 1 и 2 написаны на основании работ [231, 167], пункт 3 — на основании работы [82]. Дополнительные сведения о градиентных методах отыскания седловых точек можно найти в работах [373, 183, 334].

6.10. Метод экспоненциальных штрафов отыскания седловых точек

З а д а ч а 1. Найти точку $(x^*, y^*) \in X_0 \times Y_0$ такую, что для каждой точки (x, y) из $X_0 \times Y_0$ выполняется

$$\varphi(x^*, y) \leqslant \varphi(x^*, y^*) \leqslant \varphi(x, y^*),$$

где

$$X_0 = \{x \mid \max_{1 \leqslant i \leqslant p} g_i(x) \leqslant 0, \quad x \in \boldsymbol{R}^n\};$$

$$Y_0 = \{y \mid \max_{1 \leqslant j \leqslant m} f_j(y) \leqslant 0, \quad y \in \boldsymbol{R}^m\};$$

$$\varphi : \boldsymbol{R}^n \times \boldsymbol{R}^m \to \boldsymbol{R}^1; \quad g_i : \boldsymbol{R}^n \to \boldsymbol{R}^1, \quad i = 1, \ldots, p;$$

$$f_j : \boldsymbol{R}^m \to \boldsymbol{R}^1, \quad j = 1, \ldots, m.$$

Предположения 1. (*i*) — функция $\varphi(x, y)$ выпукла по x при каждом y и вогнута по y при каждом x; (*ii*) — $g_i(x)$, $i = 1, \ldots, p$ и f_j, $j = 1, \ldots, m$ — выпуклые функции; (*iii*) — X_0 и Y_0 — непустые ограниченные множества.

В методе экспоненциальных штрафов решение задачи 1 сводится к решению последовательности задач отыскания безусловных седловых точек (x^k, y^k) функций

$$\varphi_k(x, y) = \varphi(x, y) + \frac{1}{\alpha_k}\left[\sum_{i=1}^{p} \exp(\beta_k g_i(x)) - \sum_{j=1}^{m} \exp(\beta_k f_j(y))\right],$$

где α_k и β_k — вещественные параметры, удовлетворяющие условиям $\beta_k \to \infty$ при $k \to \infty$, $\beta_k \geqslant \alpha_k \geqslant 1$, $k \geqslant 1$.

При определенных условиях предельные точки последовательности $\{(x^k, y^k)\}_{k=1}^{\infty}$ являются решениями задачи 1. При более сильных условиях, налагаемых на функцию $\varphi(x, y)$, приводятся оценки скорости сходимости последовательности $\{(x^k, y^k)\}_{k=1}^{\infty}$ к решению (x^*, y^*) задачи 1. Удачный выбор параметров α_k и β_k получается, если $\{\alpha_k\}_{k=1}^{\infty}$, $\{\beta_k\}_{k=1}^{\infty}$, $\{\beta_k/\alpha_k\}_{k=1}^{\infty}$ стремятся к бесконечности.

Алгоритм 1

Начало. I. Задать числовые последовательности $\{\alpha_k\}_{k=1}^{\infty}$ и $\{\beta_k\}_{k=1}^{\infty}$, удовлетворяющие условиям

$$\beta_k \to \infty \quad \text{при} \quad k \to \infty;$$
$$\beta_k \geqslant \alpha_k \geqslant 1 \quad \text{при} \quad k = 1, 2, \ldots \tag{6.15}$$

(например, можно принять

$$\alpha_k = \beta_k = (\alpha)^k, \quad k = 1, 2, \ldots,$$

либо

$$\beta_k = k, \quad \alpha_k = 1, \quad k = 1, 2, \ldots,$$

либо

$$\beta_k = (\beta)^k, \quad \alpha_k = (\alpha)^k, \quad k = 1, 2, \ldots, 1 < \alpha < \beta).$$

II. Положить $k = 1$.

Основной цикл. III. Определить функцию

$$\varphi_k(x, y) = \varphi(x, y) + (1/\alpha_k) \times$$
$$\times \left[\sum_{i=1}^{p} \exp(\beta_k g_i(x)) - \sum_{j=1}^{m} \exp(\beta_k f_j(y))\right].$$

IV. Вычислить седловую точку (x^k, y^k) функции $\varphi_k(x, y)$ в $\boldsymbol{R}^n \times \boldsymbol{R}^m$, т. е. точку, для которой выполняется

$$\varphi_k(x^k, y) \leqslant \varphi_k(x^k, y^k) \leqslant \varphi_k(x, y^k), \quad \forall (x, y) \in \boldsymbol{R}^n \times \boldsymbol{R}^m.$$

V. Положить $k = k + 1$ и перейти к шагу III.

Если выполняются предположения 1 и существуют точки $x^0 \in \in \boldsymbol{R}^n$ и $y^0 \in \boldsymbol{R}^m$ такие, что

$$\lim_{\|y\| \to +\infty} \varphi(x^0, y) < +\infty; \quad \lim_{\|x\| \to +\infty} \varphi(x, y^0) > -\infty, \qquad (6.16)$$

то при каждом $k \geqslant 1$ существует, по крайней мере, одна безусловная седловая точка функции $\varphi_k(x, y)$. Если же неравенства (6.16) не выполняются, то при выполнении условия

$$\lim_{k \to \infty} (\beta_k / \alpha_k) = +\infty$$

функция $\varphi_k(x, y)$ для достаточно больших k имеет хотя бы одну седловую точку в $\boldsymbol{R}^n \times \boldsymbol{R}^m$, причем все безусловные седловые точки функции φ_k (для достаточно больших k) находятся в множестве $X_\delta \times Y_\delta$, где

$$X_\delta = \{x \mid \max_{1 \leqslant i \leqslant p} g_i(x) \leqslant \delta, \quad x \in \boldsymbol{R}^n\};$$

$$Y_\delta = \{y \mid \max_{1 \leqslant j \leqslant m} f_j(x) \leqslant \delta, \quad y \in \boldsymbol{R}^m\};$$

δ — некоторое положительное число.

Теорема 1. *Пусть выполняются предположения 1 и пусть для каждого $k \geqslant 1$ (x^k, y^k) — безусловная седловая точка функции $\varphi_k(x, y)$ в $\boldsymbol{R}^n \times \boldsymbol{R}^m$. Если последовательности $\{\alpha_k\}_{k=1}^{\infty}$ и $\{\beta_k\}_{k=1}^{\infty}$ удовлетворяют условиям (6.15) и условию*

$$\lim_{k \to \infty} (\beta_k / \alpha_k) = \infty, \qquad (6.17)$$

то каждая предельная точка (такова обязательно существует) последовательности $\{(x^k, y^k)\}_{k=1}^{\infty}$ принадлежит множеству $X_0 \times Y_0$. Кроме того, если α_k стремится к бесконечности или если множества X_0^0, Y_0^0 непусты, где

$$X_0^0 = \{x \mid g_i(x) < 0, \quad i = 1, \ldots, p, \quad x \in \boldsymbol{R}^n\}; \qquad (6.18)$$

$$Y_0^0 = \{y \mid f_j(y) < 0, \quad j = 1, \ldots, m, \quad y \in \boldsymbol{R}^m\}, \qquad (6.19)$$

то все предельные точки последовательности $\{(x^k, y^k)\}_{k=1}^{\infty}$ являются седловыми точками функции $\varphi(x, y)$ относительно $X_0 \times Y_0$, т. е. являются решением задачи 1.

Теорема 1'. *Пусть выполняются предположения 1 и условия (6.15) и (6.17). Тогда если множества X_0^0 и Y_0^0, определяемые соотношениями (6.18) и (6.19), непусты, то для достаточно больших k безусловные седловые точки (x^k, y^k) функции $\varphi_k(x, y)$ принадлежат множеству $X_0 \times Y_0$ и справедлива следующая оценка скорости сходимости по функционалу:*

$$|\varphi(x^k, y^k) - \varphi(x^*, y^*)| \leqslant (p + m)/\alpha_k.$$

Теорема 1''. *Пусть выполняются все условия теоремы 1' и пусть: (**iv**) — функция $\varphi(x, y)$ имеет единственную седловую точку (x^*, y^*)*

относительно $X_0 \times Y_0$; *(v) — функция* $\varphi(\cdot, y^*)$ *— равномерно выпукла (с функцией* $\mu_1(t)$*), а функция* $\varphi(x^*, \cdot)$ *— равномерно вогнута (с функцией* $\mu_2(t)$*). Тогда для достаточно больших* k *справедливы оценки*

$$0 \leqslant \mu_1(\| x^* - x^k \|) \leqslant (p+m)/\alpha_k;$$

$$0 \leqslant \mu_2(\| y^* - y^k \|) \leqslant (p+m)/\alpha_k.$$

Вещественно-значная функция $r(z)$ $(r: R^l \to R^1)$ называется равномерно выпуклой (соответственно, вогнутой) на выпуклом множестве Q, если существует неубывающая функция $\mu_1(t) > 0$ (соответственно, $\mu_2(t) > 0$), определенная на $(0, +\infty)$ и такая, что для всех $z, \bar{z} \in Q$, выполняется неравенство

$$r\left(\frac{z+\bar{z}}{2}\right) \leqslant \frac{1}{2} r(z) + \frac{1}{2} r(\bar{z}) - \mu_1(\| z - \bar{z} \|),$$

(соответственно,

$$r\left(\frac{z+\bar{z}}{2}\right) \geqslant \frac{1}{2} r(z) + \frac{1}{2} r(\bar{z}) + \mu_2(\| z - \bar{z} \|)).$$

Библиографические указания. При написании параграфа использовались работы [561, 527].

6.11. Задачи оптимального управления и оценивания. Методы развязывающих операторов

Задача 1. Найти такое управление $u \triangleq (u_1, u_2, \ldots, u_N) \in D \triangleq D_1 \times D_2 \times \ldots \times D_N$ для управляемой системы

$$A_k(x_k, x_{k-1}, \ldots, x_1, u_k, u_{k-1}, \ldots, u_1, z) = 0, \tag{6.20}$$

$$x \triangleq (x_1, x_2, \ldots, x_N), \quad x_k \in X_k, \quad u_k \in U_k, \quad k = 1, 2, \ldots, N,$$

с наблюдениями

$$z \triangleq (z_1, z_2, \ldots, z_N) \in Z, \; Z \triangleq Z_1 \times Z_2 \times \cdots \times Z_N,$$

$$z_k = C_k(x_k, x_{k-1}, \ldots, x_1, u_k, u_{k-1}, \ldots, u_1),$$

$$z_k \in Z_k,$$

которое обеспечит максимальное на D значение $B(x, u, z)$ (в других постановках задачи требуется выбором u обеспечить включение $B(x, u, z) \in G(u, z)$ для заданного множества $G(u, z)$) для заданной целевой функции B и заданных D, $A \triangleq (A_1, \ldots, A_N)$, $C = (C_1, \ldots, C_N)$. Предполагаем, что X_k, U_k, Z_k являются линейными (конечномерными) пространствами.

Методы развязывающих операторов основаны на использовании решения функционального уравнения [22—26]

$$\mathcal{B}(u, z, A(x, u, z), C(x, u)) = B(x, u, z), \tag{6.21}$$

определяющего функцию $\mathcal{B}_{ABCD}$ *(развязывающий оператор)*,

$$\mathcal{B}_{ABCD}(u, z) \triangleq \mathcal{B}(u, z, 0, z).$$

Если, как это бывает в реальных задачах, некоторая часть u_k^2 управления $u_k \triangleq (u_k^1, u_k^2)$ $(u \triangleq (u^1, u^2))$ является значением определенной функции U, т. е. $u^2 = U(x, u, z)$, то уравнение (6.21) принимает вид [25]

$$\mathcal{B}(u^1, U(x, u, z), z, A(x, u, z), C(x, u)) = B(x, u, z).$$

В ряде адаптивных алгоритмов оптимизации функцию $\mathcal{B}_{ABCD}$ аппроксимируют по ее вычисляемым или экспериментально определяемым на некоторой выбираемой последовательности $(u^1, z^1), \dots$ $\dots, (u^j, z^j), \dots$ значениям $B(X(u^1, z^1), u^1, z^1), \dots, B(X(u^j, z^j), u^j, z^j)$ ($X(u^j, z^j)$ — множество решений системы (6.20) при $u = u^j$, $z = z^j$).

Алгоритм 1 решения задачи 1 с помощью функции $\mathcal{B}_{ABCD}$ основан на теореме 1.

Теорема 1. *Если $\forall\ u \in D$, $z \in Z$, $X(u, z) \neq \varnothing$ и уравнение (6.21) имеет единственное решение, то решение $u(z)$ задачи параметрической оптимизации $u(z) \triangleq \arg\min_{u \in D} \mathcal{B}_{ABCD}(u, z)$ является искомым решением задачи 1.*

Приведем примеры эффективного вычисления функции $\mathcal{B}_{ABCD}$.

З а д а ч а 2. Найти управление $u \in D$ для системы

$$A_{k,k}x_k + A_{k,k-1}x_{k-1} + \cdots + A_{k,1}x_1 + f_k(u) = 0,$$
$$k = 1, 2, \dots, N,$$

которое доставляет максимальное на заданном множестве D значение для функции

$$B(x, u) \triangleq \sum_{i=1}^{N} (c_i, x_i) + f_{N+1}(u)$$

при заданных матрицах $A_{k,i}$, векторах c_i и функциях f_k.

Алгоритм 2. Вычислить решение $(y_N, y_{N-1}, \dots, y_1) \triangleq y(A, c)$ системы

$$c_k + \sum_{j=k}^{N} A_{j,k}^* y_k = 0, \quad k = N, N-1, \dots, 1, \tag{6.22}$$

и затем вычислить управление $u(y(A, c))$,

$$u(y(A, c)) \triangleq \arg\max_{u \in D} F(u, y(A, c)),$$

где

$$F(u, y) \triangleq \sum_{i=1}^{N} (y_i, f_i(u)) + f_{N+1}(u).$$

Теорема 2. *Если $y(A, c)$ — единственное решение системы (6.22) и $\forall\, u \in D$ $X(u) \neq \varnothing$, то*

$$B(X(u), u) = F(u, y(A, c))$$

и управление $u(y(A, c))$ является искомым решением задачи 2.

Следствие 1. Если $D = D_1 \times D_2 \times \cdots \times D_N$,

$$f_i(u) \equiv f_i(u_i),\ f_{N+1}(u) \equiv \sum_{i=1}^{N} \bar{f}_i(u_i),\ u_i \in D_i,$$

то управление

$$u_i(y(A, c)) \triangleq \arg\max_{u_i \in D_i} [(y(A, c), f_i(u_i)) + \bar{f}_i(u_i)]$$

является решением задачи 2.

Теорема 3. *Если*

$$f_i(u_i) = B_i u_i + d_i,\ \bar{f}_i(u_i) = (h_i, u_i),$$
$$D_i = \{u_i \mid \| u_i \|_{\nu(i)} \leqslant \alpha_i\}$$
$$(\nu(i) \in \{1, 2, 3\},\ \| u \|_1 \triangleq \max_j |(u)_j|,$$
$$\| u \|_2 \triangleq \sum_j |(u)_j|,\quad \| u \|_3 \triangleq \Bigl(\sum_j (u)_j^2\Bigr)^{1/2},$$

$(u)_j$ — j-я компонента вектора u), то управление

$$u_i(y(A, c)) \triangleq s(B_i^* y_i(A, c) + h_i, \alpha_i, \nu(i))$$

является в условиях теоремы 2 решением задачи 2, где

$(s(a, \alpha, 1))_i = \alpha \operatorname{sign}(a)_i$, $s(a, \alpha, 3) = \alpha a / \| a \|_3$, $(s(a, \alpha, 2))_i = \alpha^i(a) \operatorname{sign}(a)_i$, $\alpha^i(a)$ — *решение α^i системы*

$$\alpha^i \geqslant 0,\ \sum_i \alpha^i = \alpha,\ \forall\, i \notin \mathcal{I} \triangleq \operatorname{Arg}\max_i |(u)_i|\quad \alpha^i = 0. \tag{6.23}$$

Теорема 4. *Если векторы $u^1, \ldots, u^M$ и $a^1, \ldots, a^M$ образованы из компонент $(u_k)_j$ и $(B_k^* y_k(A, c) + h_k)_j$ таким образом, что $(u^1, \ldots, u^M) = u$, множества D_i определяются для $i \in \mathcal{I}_1$ неравенствами*

$$\| u^i \|_{\nu(i) k(i)} \leqslant \alpha_i,$$

т. е.

$$\| u^i \|_{1k} \triangleq \max_j |(k)_j (u^i)_j| \leqslant \alpha_i,\quad (k)_i > 0,$$

$$\| u^i \|_{2k} \triangleq \sum_j |(k)_j (u^i)_j| \leqslant \alpha_i,\quad \| u^i \|_{3K} \triangleq (K u^i, u^i)^{1/2} \leqslant \alpha_i,$$

K — положительно определенная матрица,

$$\| u^i \|_{(1+p)k} \triangleq \Bigl(\sum_j (k)_j |(u^i)_j|^p\Bigr)^{1/p} \leqslant \alpha_i,\ p > 0,\ p \neq 1,$$

а для $i \in \mathcal{I}_2$ — неравенствами

$$\| u^i - \omega^i \|_{\Phi_i} \leqslant \alpha_i,$$

где

$$\| u^i \|_{\Phi_i} \triangleq \min \alpha \mid \alpha^{-1} u^i \in \Phi_i,\quad \mathcal{I}_1 \cup \mathcal{I}_2 = \{1, \ldots, M\},$$

Φ_i — *выпуклые множества,*

$$F(u,\ y(A,\ c)) = \sum_{i=1}^{M} (a^i,\ u^i),$$

то в условиях теоремы 2 решение u^* *задачи 2 следующее:*
для $i \in \mathcal{I}_1$

$$u^{i*} = s_k(a^i,\ \alpha_i,\ \nu(i),\ k(i)),$$

где

$$(s(a,\ \alpha,\ 1,\ k))_j = \alpha/(k)_j \operatorname{sign}(a)_j,$$

$$s(a,\ \alpha,\ 3,\ K) = \alpha K^{-1} a \| a \|^{-1}_{3K^{-1}},$$

$$(s(a,\ \alpha,\ 2,\ k))_j = \alpha^j(a) \operatorname{sign}(a)_j,$$

$\alpha^j(a)$ — *решение* (6.23) *при*

$$\mathcal{I} = \operatorname{Arg}\max_j |(a)_j/(k)_j|,$$

$$(s(a,\ \alpha,\ 1+p,\ k))_j = \alpha(q)_j \| q \|^{-1}_{(1+p)k} \operatorname{sign}(a)_j,$$

$$(q)_j \triangleq |(a)_j/(k)_j|^{1/(p-1)};$$

для $i \in \mathcal{I}_2$

$$u^{i*} = s_k(a^i,\ \alpha_i,\ \Phi_i,\ \omega^i) \triangleq \omega^i + \alpha_i u^i_{\Phi_i},\ u^i_{\Phi_i} \in \operatorname{Arg}\max_{u \in \Phi_i} (a^i,\ u).$$

Следствие 2. В условиях теоремы 3 максимальное значение целевой функции

$$B(x(u^*),\ u^*) = F(u^*) = \sum_{i=1}^{M} \alpha_i \varphi(a_i,\ \nu(i)),$$

где $\varphi(a,\ 1) = \| a \|_2$, $\varphi(a,\ 2) = \| a \|_1$, $\varphi(a,\ 3) = \| a \|_{3K^{-1}}$, а в условиях теоремы 4

$$B(x(u^*),\ u^*) = \sum_j \alpha_j \varphi_k(a^j,\ \nu(j)) + \sum_{j \in \mathcal{I}_2} (a^j,\ \omega^j),$$

где для $i \in \mathcal{I}_1$

$$\varphi_k(a,\ 1) = \| a \|_{2k^{-1}}, \quad \varphi_k(a,\ 2) = \| a \|_{1k^{-1}},$$

$$\varphi_k(a,\ 3) = \| a \|_{3K^{-1}},\ \varphi_k(a,\ 1+p) = \| q \|^p_{(1+p)\bar{k}} \| q \|^{-1}_{(1+p)\bar{k}},\ (\bar{k})_j \triangleq (k)_j^{1/(p-1)},$$

а для $i \in \mathcal{I}_2$

$$\varphi_k(a^i,\ \nu(i)) \triangleq (a^i,\ u^i_{\Phi_i}).$$

Задача 3. Найти то управление $u = (u^1,\ \dots,\ u^M)$ (в обозначениях теорем 3, 4), которое обеспечит выполнение цели (равенства)

$$B(x(u),\ u) = b,\ b \in R^1,$$

при минимальном расходе «энергии»

$$E(u) \triangleq \sum_{i \in \mathcal{I}_1} \beta_i \| u^i \|_{\nu(i)k(i)} + \sum_{i \in \mathcal{I}_2} \beta_i \| u^i - \omega^i \|_{\Phi_i}.$$

Алгоритм 3. I. Вычисляем $y(A, c)$ (решение системы (6.22)) и затем векторы $a^1, a^2, \ldots, a^M$.

II. Вычисляем множество

$$\mathcal{I} \triangleq \operatorname{Arg} \min_{i=1,2,\ldots,M} \frac{\left(b - \sum_{i \in \mathcal{I}_2} (a^i, \omega^i)\right) \beta_i}{\varphi_k(a^i, \nu(i))}. \tag{6.24}$$

III. Вычисляем

$$u^i = s_k(a^i, \bar{\alpha}_i, \nu(i), k(i)),$$

где $\bar{\alpha}_i$ — любые числа, удовлетворяющие условиям

$$\bar{\alpha}_i \geqslant 0, \quad \forall i \notin \mathcal{I} \quad \bar{\alpha}_i = 0,$$

$$\sum_i \beta_i \bar{\alpha}_i = \min_{i=1,2,\ldots,M} \frac{\beta_i \left(b - \sum_{i \in \mathcal{I}_2} (a^i, \omega^i)\right)}{\varphi_k(a^i, \nu(i))}.$$

Задача 4. Отличается от задачи 3 наличием дополнительных ограничений

$$(\|u^i\|_{\nu(i)k(i)}, \ i \in \mathcal{I}_1; \ \|u^i - \omega^i\|_{\Phi_i}, \ i \in \mathcal{I}_2) \in Q \subset R^M. \tag{6.25}$$

Алгоритм 4. Отличается от алгоритма 3 вычислением $\bar{\alpha} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_M)$ на шаге III по формуле

$$\bar{\alpha} = \arg\min_{\bar{\alpha} \in Q} \sum_{i=1}^{M} \beta_i \bar{\alpha}_i.$$

Задача 5. Частный случай задачи 4 для

$$Q \triangleq \{u \mid \|u^i\|_{\nu(i)k(i)} \leqslant \gamma_i, \ \|u^i - \omega^i\|_{\Phi_i} \leqslant \gamma_i, \ i \in \mathcal{I}_2\}.$$

Алгоритм 5. Отличается от алгоритма 3 тем, что на шаге III вводим дополнительное ограничение $\bar{\alpha}_i \leqslant \gamma_i$. Если таких $\bar{\alpha}_i$ не найдется, то полагаем $\bar{\alpha}_i = \gamma_i$, $i \in \mathcal{I}$, заменяем число b числом $b - \sum_{i \in \mathcal{I}} \beta_i \gamma_i$, удаляем из имеющегося множества индексов i множество $\mathcal{I}$, определенное формулой (6.24), и возвращаемся на шаг II.

Теорема 5. *Если выполнены условия теоремы 2, то управления, вычисленные с помощью алгоритмов 3, 4, 5, являются решениями соответственно задач 3, 4, 5. При этом*

$$B(x(u^*), u^*) = \sum_{j=1}^{M} \beta_j \bar{\alpha}_j,$$

а для задачи 3

$$B(x(u^*), u^*) = \min_{j \in \overline{1,M}} \frac{b - \sum_{i \in \mathcal{I}_2} (a^i, \omega^i)}{\varphi_k(a^j, \nu(j))}.$$

З а д а ч а 6. Найти $u^* = \arg\min_{u \in D} E(u)$,

где

$$E(u) \triangleq \sum_{i \in \mathcal{I}_1} \beta_i \| u^i \|_{\nu(i)k(i)} + \sum_{i \in \mathcal{I}_2} \beta_i \| u^i - \omega^i \|_{\Phi_i}$$

при ограничениях (6.25) и связях

$$B_s(x(u), u) \triangleq \sum_{j=1}^{N} [(c_{sj}, x_j) + (h_{sj}, u_j)] = b_s, \quad s = 1, \ldots, S. \tag{6.26}$$

Алгоритм 6. I. Выбрать вектор $v = (v_1, v_2, \ldots, v_S)$ и вычислить

$$c_i = \sum_{s=1}^{S} v_s c_{si}, \quad h_i = \sum_{s=1}^{S} v_s h_{si},$$

$$b = \sum_{s=1}^{S} v_s b_s.$$

II. Найти (с помощью алгоритма 3) решение $u_v(y(A, c))$ задачи 3 и определить функцию $E(u_v(y(A, c)))$.

III. Найти множество V решений $v^* = (\overset{*}{v_1}, \ldots, \overset{*}{v_S})$,

$$v^* = \arg\max_{v} E(u_v(y(A, c))). \tag{6.27}$$

IV. Вычислить управление $u^* = u_v(y(A, c))$, удовлетворяющее равенствам (6.26) при $v \in V$.

Теорема 6. *Если выполнены условия теоремы 2 и множества Φ_i выпуклы, то управление, вычисленное с помощью алгоритма 6, является решением задачи 6.*

Замечание. Для ряда практически важных случаев задача (6.27) разрешается аналитически и ее решение v^* находится в явном виде.

З а д а ч а 7. Найти управление $u = u(f, k, c)$ системой

$$Ax + Bu + f = 0, \tag{6.28}$$

минимизирующее функцию

$$B(x, u) = (C_1 x, x) + (Du, x) - \frac{1}{2}(Ku, u) + (c, x) + (k, u). \tag{6.29}$$

Алгоритм 7. I. Вычисляем обратную матрицу

$$\begin{pmatrix} A + BK^{-1}D^* & BK^{-1}B^* \\ C + C^* + DK^{-1}D^* & DK^{-1}B^* + A^* \end{pmatrix}^{-1} \triangleq - \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \tag{6.30}$$

(A_{ij}, $i = 1, 2$; $j = 1, 2$ — квадратные матрицы)
и полагаем

$$X_f \triangleq A_{11}, \quad X_k \triangleq A_{11}BK^{-1} + A_{12}DK^{-1},$$

$$X_c \triangleq A_{12}, \quad U_f \triangleq K^{-1}D^*A_{11} + K^{-1}B^*A_{21},$$

$$U_k \triangleq K^{-1}D^*(A_{11}BK^{-1} + A_{12}DK^{-1}) + K^{-1}B^*(A_{21}BK^{-1} + A_{22}DK^{-1}),$$

$$U_c \triangleq K^{-1}(E + D^*A_{12} + B^*A_{22}).$$

II. Вычисляем управление

$$u(f,\ k,\ c) = U_f f + U_k k + U_c c. \qquad (6.31)$$

III. Вычисляем «траекторию»

$$x(f,\ k,\ c) = X_f f + X_k k + X_c c. \qquad (6.32)$$

IV. Вычисляем

$$B(x(f,\ k,\ c),\ u(f,\ k,\ c)) = \left(\bar{B}\begin{pmatrix} f \\ k \\ c \end{pmatrix},\ \begin{pmatrix} f \\ k \\ c \end{pmatrix}\right), \qquad (6.33)$$

где матрица $\bar{B}$ (развязывающий оператор) определяется приравниванием правых частей (6.33) и (6.29) при условиях (6.31) и (6.32).

Теорема 7. *Если существует обратная матрица (6.30), то управление $u\ (f, k, c)$ является оптимальным решением u^* задачи 7 и при этом*

$$B(x(u^*),\ u^*) = \left(\bar{B}\begin{pmatrix} f \\ k \\ c \end{pmatrix},\ \begin{pmatrix} f \\ k \\ c \end{pmatrix}\right). \qquad (6.34)$$

З а д а ч а 8 (задача минимаксной оптимизации). Найти управление $(f^*,\ k^*,\ c^*) = \arg \max\limits_{(f,k,c)\in\Phi} \min\limits_{u} B(x(u),\ u)$ для заданной целевой функции (6.29), системы (6.28) и заданного множества Φ.

Алгоритм 8. I. С помощью алгоритма 7 вычисляем матрицу $\bar{B}$.

II. Вычисляем искомые (f^*, k^*, c^*) по формуле

$$(f^*,\ k^*,\ c^*) = \arg \max_{(f,k,c)\in\Phi} \left(\bar{B}\begin{pmatrix} f \\ k \\ c \end{pmatrix},\ \begin{pmatrix} f \\ k \\ c \end{pmatrix}\right), \qquad (6.35)$$

используя для этой цели соответствующие алгоритмы, описанные выше для разных классов множеств Φ и разных способов их задания.

З а д а ч а 9 (задача оптимального оценивания). Найти линейную оценку $\hat{B}$ с минимальным уклонением для функции

$$B(G,\ H) \triangleq \sum_{ij} (p^{ij}g^{ij} + s^{ij}h^{ij})$$

и управляемой системы

$$GA_k x_k + B_k x_k + HC_k x_{k+1} + Dx_{k+1} + f_k(u) + q_k(p) = 0$$
$$k = 1,\ 2,\ \ldots,\ N,\ \ldots$$

с наблюдаемыми значениями

$$x_1,\ f_1(u),\ x_2,\ f_2(u),\ \ldots$$

в случае неизвестных элементов g^{ij} и h^{ij} матриц G и H соответственно при $ij \in \mathcal{J}_1$ и $ij \in \mathcal{J}_2$ и при неизвестном параметре p, принадлежащем заданному множеству P.

Алгоритм 9. I. Вычисляем решение $\hat{y}$ системы

$$\sum_q \sum_{k=1}^{N} y_k^i a_k^{iq} x_k^q = -p^{ij}, \quad ij \in \mathcal{J}_1,$$
$$\sum_q \sum_{k=1}^{N} y_k^i c_k^{iq} x_{k+1}^q = -s^{ij}, \quad ij \in \mathcal{J}_2, \tag{6.36}$$

определяющее развязывающий оператор для B в виде

$$B(G, H) = \mathcal{B}(y, x, f(u)) + \sum_{k=1}^{N} \sum_i y_k^i q^i(p),$$

$$\mathcal{B}(y, x, f(u)) \triangleq \sum_{k=1}^{N} \left(\sum_i y_k^i \left(f_k^i(u) + \sum_q (d_k^{iq} x_{k+1}^q + b_k^{iq} x_k^q) \right) \right) +$$
$$+ \sum_{ij \notin \mathcal{J}_1} g^{ij} \left(p^{ij} + \sum_{k=1}^{N} y_k^i \sum_\delta a_k^{i\delta} x_k^\delta \right) + \sum_{ij \notin \mathcal{J}_2} h^{ij} \left(s^{ij} + \sum_{k=1}^{N} y_k^i \sum_\delta c_k^{i\delta} x_{k+1}^\delta \right)$$

и удовлетворяющее условию

$$(\hat{p}, \hat{y}, \hat{L}) = \arg\min_{L, y \in Y} \max_{p \in P} \| r(p, y, L) \|, \; r(p, y, L) \triangleq L + \sum_{k=1}^{N} \sum_i y_k^i q_k^i(p). \tag{6.37}$$

II. Вычисляем оценку $\hat{B}$ в виде

$$\hat{B}(\hat{y}, \hat{L}', x, f(u)) = B(\hat{y}, x, f(u)) - \hat{L}$$

с минимальным на Y уклонением $\| r(p, y, L) \|$.

Теорема 8.

$$B(G, H) - \hat{B}(\hat{y}, \hat{L}, x, f(u)) + \hat{L} = r(p, \hat{y}, \hat{L}). \tag{6.38}$$

Задача 10 (задача наблюдения и гарантированного (минимаксного) управления). Пусть $\bar{x}_k \triangleq (x_1, x_2, \ldots, x_k)$, $\bar{u}_k \triangleq (u_1, u_2, \ldots, u_k)$, $\bar{z}_k \triangleq (z_1, z_2, \ldots, z_k)$ и управляемая система описывается уравнениями

$$A_k(\bar{x}_k, \bar{u}_k, \bar{z}_k, p) = 0, \quad k = 1, 2, \ldots, \tag{6.39}$$

где p — вектор неизвестных параметров системы, $\bar{z}_k$ — вектор наблюдений на k-м этапе, удовлетворяющий равенствам

$$C_k(\bar{x}_k, \bar{u}_k, \bar{z}_k, p) = 0, \quad k = 1, 2, \ldots,$$

а $\bar{u}_k$ — вектор управляющих воздействий (управлений). Требуется найти гарантирующее управление

$$u_k \triangleq u_k(P_k, \bar{z}_k) \triangleq \arg\min_{u \in \Phi} \max_{p \in P_k} B(\bar{x}_N, (\bar{u}_N \; \bar{z}_k, p), \bar{u}_N, \bar{z}_k, p) \tag{6.40}$$

с помощью множеств

$$P_k \triangleq \{p \mid F_s(p) \leqslant \alpha_s, \; s \in S(P_{k-1})\} \cap P_{k-1},$$

где

$$\alpha_s = \{ \max_{p \in P_{k-1}} F_s(p) \mid C_j(\bar{x}_j, \bar{u}_j, \bar{z}_j, p) = 0, \quad j = 1, 2, \ldots, k\}. \quad (6.41)$$

Замечание. Множества $S(P_{k-1})$, как и функции F_s, обычно выбирают на k-м этапе неформальными методами, преследуя при этом несколько противоречивые требования: уменьшить, с одной стороны, трудоемкость решения вспомогательных задач (6.40) и (6.41), а с другой — обеспечить «эффективное» сужение множества неопределенности P_k, содержащего неизвестный параметр p. В качестве функций F_s часто удобно выбирать линейные функции. Для решения сформулированных (под) задач (6.40) и (6.41) можно использовать описанные выше алгоритмы максиминной оптимизации. В частности, для управляемой системы (6.28) и функционала (6.29) при $f = f(p)$, $k = k(p)$ и $c = c(p)$ для решения задачи (6.40) можно использовать алгоритм 10.

Алгоритм 10. Вычисляем матрицы U_f, U_k, U_c, $\bar{B}$ с помощью алгоритма 7 и согласно (6.31) вычисляем решение задачи (6.40) по формуле

$$u_k(P_k, \bar{z}_k) = U_f f(p^*) + U_k k(p^*) + U_c c(p^*),$$

$$p^* = \arg\max_{p \in P_k} \left(\bar{B} \begin{pmatrix} f(p) \\ k(p) \\ c(p) \end{pmatrix}, \begin{pmatrix} f(p) \\ k(p) \\ c(p) \end{pmatrix} \right).$$

В общем случае для решения задачи (6.40) можно использовать методы *асимптотически развязывающих операторов*, которые основаны на локальной аппроксимации развязывающего оператора в окрестности заданной точки $\hat{u}$ функцией $B_{s,\hat{u}}$ (асимптотически развязывающим оператором s-го порядка), удовлетворяющей условию

$$B_{s,\hat{u}}(u, \bar{z}_N) = B(\bar{x}_N(u, \bar{z}_N), u, \bar{z}_N, p) + o(\rho^s(u, \hat{u})),$$

(ρ — метрика в пространстве управлений).

Алгоритм 11. I. Выбираем $q = 1$, произвольные числа $\alpha_1, \alpha_2 > 0$, α_3, $\alpha_4 > 0$, произвольный элемент $\hat{u}_q \in \Phi$.

II. Выбираем некоторую реализуемую аппроксимацию B_σ, $\sigma \in [1, s]$, функции $B_{s,\hat{u}_q}$ и вычисляем элемент $\hat{u}_{q+1} \in \Phi$ по формуле

$$\hat{u}_{q+1} = \arg \min_{u \in E(\lambda_q, \hat{u}_q)} B_\sigma(u),$$

где

$$E(\lambda_q, \hat{u}_q) \triangleq \{u \mid \rho(u, \hat{u}_q) \in [\lambda_q, \alpha_3\lambda_q], u \in \Phi\},$$

$\lambda_q = 2^{-N}\alpha_4\lambda_{q-1}$, N — минимальное натуральное число, удовлетворяющее неравенству

$$B(\bar{x}_N(\hat{u}_{q+1}, \bar{z}_N), \hat{u}_{q+1}, \bar{z}_N, p) \leqslant$$
$$\leqslant B(\bar{x}_N(\hat{u}_q, \bar{z}_N), \hat{u}_q, \bar{z}_N, p) - \alpha_1\lambda_q^{\sigma+1}.$$

Теорема 9. *Если* $\forall u \in \Phi$

$$B(\bar{x}_N(u, \bar{z}_N), u, z_N) \geqslant \alpha_5 > -\infty,$$
$$|B_\sigma(u, \bar{z}_N) - B_{s,\hat{u}_q}(\bar{u}, \bar{z}_N)| \leqslant \alpha_6\rho^\mu(u, \bar{u}),$$

то для любого $\varepsilon > 0$ *за конечное число итераций* $q = q(\varepsilon, \mu, \alpha_5, \alpha_6)$ *будет вычислено* ε*-экстремальное решение* u_ε μ*-го порядка* (элемент $u_\varepsilon \in \Phi$ называем ε-экстремальным решением μ-го порядка для задачи (6.40), если не существует ветви роста μ-го порядка, т. е. последовательности $\{u_j\}_{j=1}^{\infty} \in \Phi$, удовлетворяющей условиям

$$u_j \neq u_\varepsilon, \ \rho(u_j, u_\varepsilon) \to 0, \ \sum_{i=1}^{\mu} \nu_i\rho^i(u_j, u_\varepsilon) \geqslant \varepsilon\rho^\mu(u_j, u_\varepsilon),$$

$$\sum_{i=0}^{\mu} \nu_i\rho^{i-\mu}(u_j, u_\varepsilon) \to 0, \ \nu_0 = B(\bar{x}_N(u_\varepsilon, \bar{z}_N), u_\varepsilon, \bar{z}_N) -$$
$$- B(\bar{x}_N(u_j, \bar{z}_N), u_j, \bar{z}_N)).$$

Способ вычисления требуемой функции $B_{1,\hat{u}_q}$ для весьма общего случая $B(x, u, z) = \mathrm{Blg}(\bar{B}(x, u, z))$ $(\bar{B} = (\bar{B}_1, \ldots, \bar{B}_L)$, $\mathrm{Blg}: \mathbf{R}^L \to \mathbf{R}^1$ — заданная функция с логическими операциями, включая операции max и min) дает теорема 10.

Теорема 10. *Если для* $A_{ki} \triangleq \partial_{x_i}A_k$, $\overset{s}{c_i} \triangleq \partial_{x_i}\bar{B}_s$ *в точке* $(x(\hat{u}_q, \bar{z}_N), \hat{u}_q, \bar{z}_N)$ *решение* $y(A, c^s)$ $(c^s \triangleq (\overset{s}{c_1}, \ldots, \overset{s}{c_N}))$ *системы* (6.22) *единственно и функции* A_k, $\bar{B}_s$, $\partial_{x_i}A_k$, $\partial_{x_i}\bar{B}_s$ *непрерывны по* x, u, *то* $B_{1,\hat{u}_q}(u, \bar{z}_N) = \mathrm{Blg}(\hat{B}(u, \bar{z}_N))$, *где*

$$\hat{B}_s(u, \bar{z}_N) \triangleq \bar{B}_s(x(\hat{u}_q, \bar{z}_N), u, \bar{z}_N) + \sum_{k=1}^{N} (y_k(A, c^s), A_k(x(\hat{u}_q, \bar{z}_N), u, \bar{z}_N)).$$

Основные трудности реализации алгоритма 11 часто связаны с трудностями вычисления значений $B(\bar{x}_N(u_q, \bar{z}_N), u_q, \bar{z}_N)$ для заданных управлений u_q. В таких случаях можно эффективно использовать метод *развязывающей декомпозиции* для упрощения математической модели $A(x, u) = 0$ с помощью выбора таких функций

$$A_1, A_2, A_3, A(x, u) \equiv (A_1(x_1, A_2(x, u), u), A_3(x, u)), \ x \triangleq (x_1, x_2),$$

для которых упрощается решение задачи $x_1(v, u) = \arg_z [A_1(z, v, u) = 0]$. Это дает возможность искать u^* как решение задачи минимизации функции

$$B_1(x_1, v, u) \triangleq B(x, u, z) + K_1 \| v - A_2(x, u) \| + K_2 \| A_3(x, u) \|,$$

$$K_1, K_2 \in R^1_+,$$

при упрощенных связях $A_1(x_1, v, u) = 0$ (или вычислять $x_2(u)$ решением вспомогательной задачи

$$(x_2(u), v) = \arg [A_2(x_1(u), x_2, u) = v, A_3(x_1(u), x_2, u) = 0]).$$

Библиографические указания. При написании параграфа использованы результаты работ [22—26], [39—41], [135].

СПИСОК ЛИТЕРАТУРЫ

1. *Абрамов А. П., Иванилов Ю. П.* Алгоритм решения задачи линейного программирования методом нагруженного функционала.— Журн. вычисл. математики и мат. физики, 1977, № 1, т. 17, с. 259—262.
2. *Айда-Заде К. Р., Махмуд-Заде Р. Н., Новрузбеков И. Г.* Метод полярных координат минимизации функций овражной структуры.— В кн.: Численные методы нелинейного программирования: Тез. II Всесоюз. семинара. Харьков, 28 мая — 3 июня 1976 г. Харьков : Вища школа. Изд-во при Харьк. ун-те, 1976, с. 30—32.
3. *Айзерман М. А., Браверман Э. М., Розоноэр Л. И.* Метод потенциальных функций в теории обучения машин.— М. : Наука, 1970.— 384 с.
4. *Алгоритмы* и программы случайного поиска / Под ред. Л. А. Растригина.— Рига: Зинатне, 1969.— 374 с.
5. *Амвросиенко В. В.* Ускорение сходимости метода Брауна решения матричных игр.— Экономика и мат. методы, 1965, т. 1, № 4, с. 570—575.
6. *Антонов Г. Е., Катковник В. Я.* Метод синтеза одного класса алгоритмов случайного поиска.— Автоматика и телемеханика, 1971, № 6, с. 154—157.
7. *Антипин А. С.* О методе выпуклого программирования, использующем симметрическую модификацию функции Лагранжа.— Экономика и мат. методы, 1976, т. 12, № 6, с. 1164—1173.
8. *Арбузова Н. И., Launch Вересков А. И., Николаева Н. Д.* Некоторые задачи стохастического программирования (обзор).— Экономика и мат. методы, 1969, т. 5, № 3, с. 412—430.
9. *Афанасьев А. Ю.* О поиске минимума функции с ограниченной второй производной.— Журн. вычисл. математики и мат. физики, 1974, т. 14, № 4, с. 1018—1021.
10. *Афанасьев А. Ю., Новиков В. А.* О поиске минимума функции с ограниченной третьей производной.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 4, с. 1031—1034.
11. *Ахметов П. А., Малков У. Х.* Повышение эффективности мультипликативного алгоритма симплекс-метода при решении больших задач линейного программирования на ЭВМ.— Экономика и мат. методы, 1970, т. 6, вып. 3, с. 422—426.
12. *Бабич М. Д., Иванов В. В.* Исследование полной погрешности в задачах минимизации функционалов при наличии ограничений.— Укр. мат. журн., 1969, т. 21, № 1, с. 3—14.
13. *Баженов Л. Г.* Об условиях сходимости метода минимизации почти дифференцируемых функций.— Кибернетика, 1972, № 4, с. 71—72.
14. *Баженов Л. Г., Гупал А. М.* Об одном стохастическом аналоге метода возможных направлений.— Кибернетика, 1973, № 4, с. 94—95.
15. *Баничук Н. В., Петров В. М., Черноусько Ф. Л.* Численное решение вариационных и краевых задач методом локальных вариаций.— Журн. вычисл. математики и мат. физики, 1966, т. 6, № 6, с. 947—961.
16. *Бартиш М. Я.* Возмущенные аналоги методов типа Ньютона — Канторовича. Матем. сб.— Киев : Наук. думка, 1976, с. 59—62.
17. *Батищев Д. И.* Поисковые методы оптимального проектирования.— М. : Сов. радио, 1975.— 216 с.

18. *Батищев Д. И., Бедная Р. И., Стронгин Р. Г.* О выборе параметров алгоритмов поисковой оптимизации.— Автоматика и вычисл. техника, 1972, № 4, с. 56—60.
19. *Бахвалов Н. С.* О свойствах оптимальных методов решения задач математической физики.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 3, с. 555—568.
20. *Бахтин А. Е.* Блочный способ решения задач линейного программирования.— В кн. : Математические методы решения экономических задач. Новосибирск: Наука, 1971, с. 63—78.
21. *Бахтин А. Е., Волков Ю. И.* Метод последовательного улучшения плана для решения одного класса задач линейного программирования.— Экономика и мат. методы, 1967, т. 3, вып. 4, с. 581—587.
22. *Бейко И. В.* Применение развязывающих операторов для решения задачи Коши и построения экстремальных алгоритмов развязывающей декомпозиции. Вычислиг. и прикл. математика, 1981, вып. 45, с. 110—118.
23. *Бейко И. В.* Экстремальные модели для численного исследования оптимальных процессов в задачах теплопроводности при наличии погрешностей в исходной информации.— В кн.: Аналитические, численные и аналоговые методы в задачах теплопроводности. Киев : Наук. думка, 1977, с. 220—233.
24. *Бейко И. В.* Численный анализ графооператорных уравнений методом развязывающих операторов и *s*-экстремальных моделей.— В кн.: Вычислительная математика в современном научно-техническом прогрессе. Киев : Вища школа. Изд-во при Киев. ун-те, 1978, с. 124—125.
25. *Бейко И. В.* Экстремальные модели сложных систем и метод декомпозиции в вычислительном эксперименте.— Вестн. Киев. ун-та. Моделирование и оптимизация систем, 1982, вып. 1, с. 72—81.
26. *Бейко И. В., Бейко М. Ф.* Новый подход к математическому моделированию и исследованию сложных процессов.— В кн.: Задачи гидроаэромеханики. Киев : Изд. Ин-та математики АН УССР, 1973, с. 229—236.
27. *Бейко И. В., Бейко М. Ф.* К численному построению оптимальных управлений.— В кн.: Нестационарные процессы. Киев : Изд. Ин-та математики АН УССР, 1977, с. 174—190.
28. *Бейко И. В., Бейко М. Ф.* Численные методы решения задач оптимального управления.— Киев : Знание, 1963.— 141 с.
29. *Бейко И. В., Зинько П. Н.* Некоторые вероятностные оценки для алгоритмов стохастического программирования с постоянным шаговым множителем.— Кибернетика, 1978, № 2, с. 91—95.
30. *Бейко И. В., Зинько П. Н.* Применение метода стохастического градиента к решению некоторых минимаксных задач управления.— В кн.: Научная конференция «Вычислительная математика в современном научно-техническом прогрессе». Канев, 1974, вып. 1, с. 12—18.
31. *Бейко И. В., Зинько П. Н.* Применение метода стохастических градиентов к решению стохастических задач уклонения.— В кн.: Моделирование и оптимизация систем управления. Киев : Вища школа. Изд-во при Киев. ун-те, 1974, с. 34—42.
32. *Бейко И. В., Зинько П. Н.* Численные методы решения непрерывной минимаксной задачи стохастического программирования.— В кн.: Численные методы нелинейного программирования: Тез. II Всесоюзного семинара, Харьков, 31 мая — 3 июня, 1976, Харьков : Вища школа. Изд-во при Харьк. ун-те, 1976, с. 69—74.
33. *Бейко И. В., Ясинский В. В.* Задачи принятия решений в конфликтных ситуациях. Пр. 77—27. Киев : Изд. Ин-та математики АН УССР, 1977.— 48 с.
34. *Беленький В. З., Волконский В. А., Иванков С. А., Поманский А. Б., Шапиро А. Д.* Итеративные методы в теории игр и программировании.— М. : Наука, 1974.— 240 с.
35. *Белов Е. Н.* Алгоритм решения задач квадратичного и линейного программирования методом сопряженных градиентов с использованием модифицированных функций Лагранжа (Алгол).— Программы и алгоритмы, 1974, вып. 57, с. 10—13.

36. *Бирзак Б., Пшеничный Б. Н.* О некоторых задачах минимизации недифференцируемых функций.— Кибернетика, 1966, № 6, с. 53—57.
37. *Богомолов Н. А., Карманов В. Г.* О методе вычисления стационарных точек общей задачи нелинейного программирования.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 1, с. 72—78.
38. *Бублик Б. Н.* Численные решения динамических задач теории пластин и оболочек.— Киев : Наук. думка, 1976.— 376 с.
39. *Бублик Б. Н., Бейко И. В., Зинько П. Н., Харченко А. С.* Об адаптивном управлении системами переменной структуры при наличии возмущений.— В кн.: Всесоюз. конф. по оптимальному управлению в механических системах. Казань, 1978, с. 5—7.
40. *Бублик Б. Н., Кириченко Н. Ф.* О разностном подходе к решению задач управления систем с распределенными параметрами.— Численные методы механики сплошной среды, 1977, т. 8, № 3, с. 9—11.
41. *Бублик Б. Н., Кириченко Н. Ф., Наконечный А. Г.* Минимаксные оценки и регуляторы в динамических системах.— Пр. ИК АН УССР, Киев, 1978.— 48 с.
42. *Бублик Б. Н., Кириченко Н. Ф., Наконечный А. Г.* Регуляторы и минимаксные фильтры для систем с распределенными параметрами.— Пр. ИК АН УССР, Киев, 1978.— 40 с.
43. *Булавский В. А.* О разложении квадратичной матрицы в произведение ортогональной и треугольной.— Сибир. мат. журн., 1969, т. 10, № 2, с. 467—469
44. *Булавский В. А.* Итеративный метод решения общей задачи линейного программирования.— В кн.: Численные методы оптимального планирования. Новосибирск : Изд-во АН СССР, 1962, вып. 1, с. 35—64.
45. *Булавский В. А., Звягина Р. А., Яковлева М. А.* Численные методы линейного программирования / Под ред. Канторовича Л. В.— М. : Наука, 1977.—368 с.
46. *Булатов В. П.* Методы аппроксимации при решении некоторых экстремальных задач: Автореф. дис. ... канд. физ.-мат. наук.— Томск, 1967 — 16 с.
47. *Бусленко Н. П., Шрейдер Ю. А.* Метод статистических испытаний (Монте-Карло) и его реализация.— М. : Физматгиз, 1961.— 226 с.
48. *Васильев Л. В., Митчел Б. Ф.* Один конечный метод нахождения направления наискорейшего спуска.— Оптимизация, 1973, вып. 10, с. 59—72.
49 *Васильев Ф. П.* Лекции по методам решения экстремальных задач.— М. : Изд-во МГУ, 1974.— 374 с.
50. *Ватель И. А., Ерешко Ф. И.* Математика конфликта и сотрудничества.— М. : Знание, 1973.— 64 с.
51. *Венец В. И.* Седловая точка функции Лагранжа и негладкие штрафные функции в выпуклом программировании.— Автоматика и телемеханика, 1974, № 8, с. 109—118.
52. *Верина Л. Ф., Танаева В. С.* Декомпозиционные подходы к решению задач математического программирования (обзор).— Экономика и мат. методы, 1975, т. 11, вып. 6, с. 1160—1172.
53. *Верченко П. И.* Об одном непоисковом алгоритме адаптивного управления.— Кибернетика, 1976, № 1, с. 136—139.
54. *Волкович В. Л., Волошин А. Ф.* Об одной схеме метода последовательного анализа и отсеивания вариантов.— Кибернетика, 1978, № 4, с. 98—105.
55. *Волконский В. А.* Оптимальное планирование в условиях большой размерности (итеративные методы и принцип декомпозиции).— Экономика и мат. методы, 1965, т. 1, вып. 2, с. 195—220.
56. *Волынский Э. И.* Ослабленная задача нелинейного программирования и метод штрафных оценок.— Кибернетика, 1977, № 5, с. 103—107.
57. *Волынский Э. И., Почтман Ю. М.* Об одном алгоритме случайного поиска для решения многоэкстремальных задач.— Изв. АН СССР. Техн. кибернетика, 1974, № 1, с. 55—60.
58. *Вопросы* теории и элементы программного обеспечения минимаксных задач / Под ред. В. Ф. Демьянова, В. Н. Малоземова.— Л. : Изд-во ЛГУ, 1977.— 192 с.
59. *Габасов Р., Кириллова Ф. М.* Методы линейного программирования. Общие задачи.— Минск : Изд-во Белорус. ун-та, 1977.— 175 с.

60. *Габасов Р., Кириллова Ф. М.* Методы оптимизации.— Минск : Изд-во Белорус. ун-та, 1975.— 279 с.
61. *Гантмахер Ф. Р.* Теория матриц.— М. : Гостехиздат, 1953.— 492 с.
62. *Ганшин Г. С.* Расширение области сходимости метода Ньютона.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 5, с. 1294—1296.
63. *Ганшин Г. С.* Вычисление наибольшего значения функций.— Журн. вычисл. математики и мат. физики, 1976, т. 16, № 1, с. 30—39.
64. *Гасс С.* Линейное программирование.— М. : Физматгиз, 1961.— 304 с.
65. *Гасс С.* Первое допустимое решение в задачах линейного программирования.— В кн.: Методы решения общей задачи линейного программирования. М. : Госстатиздат, 1963, с. 47—54.
66. *Гермейер Ю. Б.* Необходимые условия максимина.— Журн. вычисл. математики и мат. физики, 1969, т. 9, № 2, с. 432—438.
67. *Гермейер Ю. Б.* Приближенное сведение с помощью штрафных функций задачи определения максимина к задаче определения максимума.— Журн. вычисл. математика и мат. физики, 1969, т. 9, № 3, с. 730—731.
68. *Гермейер Ю. Б.* К задаче отыскания максимина с ограничениями.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 1, с. 39—54.
69. *Гермейер Ю. Б.* Введение в теорию исследования операций.— М. : Наука, 1971.— 384 с.
70. *Гермейер Ю. Б.* Игры с непротивоположными интересами.— М. : Наука, 1976.— 328 с.
71. *Гермейер Ю. Б., Ерешко Ф. И.* Побочные платежи в играх с фиксированной последовательностью ходов.— Журн. вычисл. математики и мат. физики, 1974, т. 14, № 6, с. 1437—1450.
72. *Гермейер Ю. Б., Крылов И. А.* Поиск максиминов методом «невязок».— Журн. вычисл. математики и мат. физики, 1972, т. 12, № 4, с. 871—881.
73. *Гирсанов И. В.* Лекции по математической теории экстремальных задач.— М. : Изд-во МГУ, 1970.— 118 с.
74. *Глушков В. М.* О диалоговом методе решения оптимизационных задач.— Кибернетика, 1975, № 4, с. 2—6.
75. *Глушков В. М.* Некоторые задачи создания и развития методов и средств кибернетики и вычислительной техники, стоящие перед молодыми учеными института кибернетики АН УССР.— Кибернетика, 1978, № 5, с. 3—5.
76. *Глушков В. М., Кудринский В. Ю., Охрименко М. Г.* Об одной модели планирования на макроуровне с оптимизацией транспортных затрат.— Докл. АН СССР, 1976, т. 231, № 3, с. 355—358.
77. *Голиков А. И., Евтушенко Ю. Г.* Об одном классе методов решения задач нелинейного программирования.— Докл. АН СССР, 1978, т. 239, № 3, с. 519—522.
78. *Гольштейн Е. Г.* Об одном видоизменении метода последовательного улучшения плана.— Экономика и мат. методы, 1965, т. 1, вып. 1, с.. 83—85.
79. *Гольштейн Е. Г.* Методы блочного программирования.— Экономика и мат. методы, 1966, т. 2, вып. 1, с. 82—114.
80. *Гольштейн Е. Г.* Выпуклое программирование. Элементы теории.— М. : Наука, 1970.— 68 с.
81. *Гольштейн Е. Г.* Теория двойственности в математическом программировании и ее приложения.— М. : Наука, 1971.— 352 с.
82. *Гольштейн Е. Г.* Обобщенный градиентный метод отыскания седловых точек.— Экономика и мат. методы, 1972, т. 8, вып. 4, с. 569—580.
83. *Гольштейн Е. Г.* О сходимости градиентного метода отыскания седловых точек модифицированных функций Лагранжа.— Экономика и мат. методы, 1977, т. 13, вып. 2, с. 322—329.
84. *Гольштейн Е. Г., Буй Тхе Там.* Об одном итерационном методе выпуклого программирования, использующем модифицированные функции Лагранжа.— Экономика и мат. методы, 1977, т. 13, вып. 6, с. 1271—1278.
85. *Гольштейн Е. Г., Мовшович С. М.* Современные направления в математическом программировании.— Экономика и мат. методы, 1967, т. 3, вып. 5, с. 766—778.

86. *Гольштейн Е. Г., Третьяков Н. В.* Модифицированные функции Лагранжа.— Экономика и мат. методы, 1974, т. 10, вып. 3, с. 568—591.
87. *Гольштейн Е. Г., Третьяков Н. В.* Градиентный метод минимизации и алгоритмы выпуклого программирования, связанные с модифицированными функциями Лагранжа.— Экономика и мат. методы, 1975, т. 11, вып. 4, с. 730—742.
88. *Гольштейн Е. Г., Юдин Д. Б.* Новые направления в линейном программировании.— М. : Сов. радио, 1966.— 524 с.
89. *Горский А. А.* Модифицированные методы штрафных функций для решения задач выпуклого программирования.— Изв. АН СССР. Техн. кибернетика, 1971, № 6, с. 25—29.
90. *Грачев Н. И., Евтушенко Ю. Г.* Библиотека программ для решения задач оптимального управления.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 2, с. 367—387.
91. *Гупал А. М.* Метод оптимизации в нестационарных условиях.— Кибернетика, 1974, № 3, с. 131—133.
92. *Гупал А. М.* Прямой метод решения задач стохастического программирования.— Автоматика и телемеханика, 1977, № 4, с. 61—65.
93. *Гупал А. М.* Об одном методе минимизации почти дифференцируемых функций.— Кибернетика, 1977, № 1, с. 114—116.
94. *Гупал А. М.* Об одной задаче стохастического программирования с ограничениями вероятностной природы.— Кибернетика, 1974, № 6, с. 94—100.
95. *Гупал А. М.* Об аналоге метода возможных направлений в задачах минимизации недифференцируемых функций.— Кибернетика, 1978, № 2, с. 62—64.
96. *Гупал А. М.* Методы минимизации функций, удовлетворяющих условию Липшица, с усреднением направлений спуска.— Кибернетика, 1978, № 5, с. 49—51.
97. *Гупал А. М.* Стохастические методы решения негладких экстремальных задач.— Киев : Наук. думка, 1979.— 152 с.
98. *Гупал А. М., Баженов Л. Г.* Стохастический аналог метода сопряженных градиентов.— Кибернетика, 1972, № 1, с. 125—126.
99. *Гупал А. М., Баженов Л. Г.* Стохастический метод линеаризации.— Кибернетика, 1972, № 3, с. 116—117.
100. *Гупал А. М., Голодников А. Н.* О свойствах класса почти дифференцируемых функций.— В кн.: Теория оптимальных решений. Киев : Изд-во Ин-та кибернетики АН УССР, 1976, с. 21—28.
101. *Гупал А. М., Голодников А. Н.* Некоторые методы решения предельных экстремальных задач.— В кн.: Методы исследования операций и теории надежности в анализе систем. Киев : 1976, с. 66—74.
102. *Гупал А. М., Мирзоахмедов Ф.* Об одном способе регулировки шага в методах стохастического программирования.— Кибернетика, 1978, № 1, с. 133—134.
103 *Гупал А. М., Норкин В. И.* Алгоритм минимизации разрывных функций.— Кибернетика, 1977, № 2, с. 73—75.
104. *Гурин Л. С., Лобач В. П.* Комбинация метода Монте-Карло с методом скорейшего спуска при решении некоторых экстремальных задач.— Журн. вычисл. математики и мат. физики, 1962, т. 2, № 3, с. 499—502.
105. *Данилин Ю. М.* Об одном подходе к задачам минимизации.— Докл. АН СССР, 1969, т. 188, № 6, с. 1221—1222.
106. *Данилин Ю. М.* Методы минимизации, основанные на аппроксимации исходного функционала выпуклым.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 5, с. 1067—1080.
107. *Данилин Ю. М.* Оценка эффективности одного алгоритма отыскания абсолютного минимума.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 4, с. 1026—1031.
108. *Данилин Ю. М.* Методы сопряженных направлений для решения задач минимизации.— Кибернетика, 1971, № 5, с. 122—136.
109. *Данилин Ю. М.* Скорость сходимости методов сопряженных направлений.— Кибернетика, 1977, № 6, с. 97—105.
110. *Данилин Ю. М., Панин В. М.* Методы сопряженных направлений без вычис-

ления производных.— В кн.: Численные методы нелинейного программирования. Киев : Наук. думка, 1976, с. 104—107.
111. *Данилин Ю. М., Пшеничный Б. Н.* О методах минимизации с ускоренной сходимостью.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 6, с. 1341—1354.
112. *Данилин Ю. М., Пшеничный Б. Н.* Метод минимизации без вычислений производных.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 1, с. 12—21.
113. *Данскин Дж. М.* Теория максимина и ее приложения к задачам распределения вооружения.— М. : Сов. радио, 1970.— 200 с.
114. *Данциг Дж.* Линейное программирование, его обобщения и применения.— М. : Прогресс, 1966.— 600 с.
115. *Данциг Дж., Вольф Ф.* Алгоритм разложения для задач линейного программирования.— Математика, 1964, т. 8, № 1, с. 151—160.
116. *Данциг Дж., Форд Л., Фулдерсон Д.* Алгоритм для одновременного решения прямой и двойственной задач линейного программирования.— В кн.: Линейные неравенства и смежные вопросы / Под ред. Г. Куна и А. Таккера. 1959, М. : Изд-во иностр. лит. с. 277—286.
117. *Демьянов В. Ф.* К решению некоторых минимаксных задач, I.— Кибернетика, 1966, № 6, с. 58—66.
118. *Демьянов В. Ф.* К решению некоторых минимаксных задач, II.— Кибернетика, 1967, № 3, с. 62—66.
119. *Демьянов В. Ф.* Дифференцирование функций максимина, I.—Журн. вычисл. математики и мат. физики, 1968, т. 8, № 6, с. 1186—1195.
120. *Демьянов В. Ф.* К задаче о минимаксе.— Докл. АН СССР, 1969, т. 187, № 2, с. 255—258.
121 *ДемьяновВ. Ф.* К отысканию минимакса на ограниченном множестве.— Докл. АН СССР, 1970, т. 191, № 6, с. 1216—1219.
122. *Демьянов В. Ф.* Дифференцирование функции максимина, II.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 1, с. 26—38.
123. *Демьянов В. Ф.* Ускорение сходимости при решении минимаксных задач.— Журн. вычисл. математики и мат. физики, 1972, т. 12, № 1, с. 51—60.
124. *Демьянов В. Ф.* К методу экстремального базиса.— Докл. АН СССР, 1976, т. 229, № 2, с. 272—275.
125. *Демьянов В. Ф.* Метод экстремального базиса в минимаксных задачах.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 2, с. 512—517.
126. *Демьянов В. Ф., Васильев Л. В.* Недифференцируемая оптимизация.— М. : Наука, 1981, 384 с.
127. *Демьянов В. Ф., Малоземов В. Н.* Введение в минимакс.— М. : Наука, 1972.— 368 с.
128. *Демьянов В. Ф., Малоземов В. Н.* К теории нелинейных задач.— Успехи мат. наук, 1971, т. 26, № 3, с. 53—104.
129. *Демьянов В. Ф., Полякова Л. Н., Рубинов А. М.* Об одном обобщении понятия субдифференциала.— В кн.: Тезисы Всесоюзной конференции по динамическому управлению.— Свердловск, 1979, с. 79—84.
130. *Демьянов В. Ф., Рубинов А. М.* Приближенные методы решения экстремальных задач.— Л. : Изд-во ЛГУ, 1968.— 180 с.
131. *Денисов Д. В.* Метод покоординатного спуска в задачах условной оптимизации.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 4, с. 1034—1036.
132. *Денисов Д. В.* О методе случайного поиска в задачах условной минимизации.— Журн. вычисл. математики и мат. физики, 1978, т. 18, № 5, с. 1103—1111.
133. *Дубовицкий А. Я., Милютин А. А.* Задачи на экстремум при наличии ограничений.— Журн. вычисл. математики и мат. физики, 1965, т. 5, № 3, с. 395—453.
134. *Евтушенко Ю. Г.* Численный метод отыскания наилучших гарантирующих оценок.— Журн. вычисл. математики и мат. физики, 1972, т. 12, № 1, с. 89—104.

135. *Евтушенко Ю. Г.* Методы решения экстремальных задач и их применение в системах оптимизации.— М.: Наука, 1982.—432 с..
136. *Евтушенко Ю. Г.* Численные методы решения задач нелинейного программирования.— Журн. вычисл. математики и мат. физики, 1976, т. 16, № 2, с. 307—324.
137. *Евтушенко Ю. Г.* Численный метод поиска глобального экстремума (перебор на неравномерной сетке).— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 6, с. 1390—1403.
138. *Евтушенко Ю. Г., Жадан В. Г.* Применение метода функций Ляпунова для исследования сходимости численных методов.— Журн. вычисл. математики и мат. физики, 1975, т. 15, № 1, с. 101—112.
139. *Евтушенко Ю. Г., Жадан В. Г.* Релаксационный метод решения задач нелинейного программирования.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 4, с. 890—904.
140. *Еремин И. И.* О некоторых итерационных методах в выпуклом программировании.— Экономика и мат. методы, 1966, т. 2, № 6, с. 870—886.
141. *Еремин И. И.* О методе штрафов в выпуклом программировании.— Кибернетика, 1967, № 4, с. 63—67.
142. *Еремин И. И.* Методы фейеровских приближений в выпуклом программировании.— Мат. заметки, 1968, т. 3, № 2, с. 217—234.
143. *Еремин И. И.* Применение метода фейеровских приближений к решению задач выпуклого программирования с негладкими ограничениями.— Журн. вычисл. математики и мат. физики, 1969, т. 9, № 5, с. 1153—1160.
144. *Еремин И. И.* Фейеровские отображения и задача выпуклого программирования.— Сиб. мат. журн., 1969, т. 10, № 5, с. 1034—1047.
145. *Еремин И. И.* Дискретные процессы фейеровского типа для негладких задач выпуклого программирования.— Алгебра и логика, 1976, т. 15, № 6, с. 628—641.
146. *Еремин И. И., Астафьев Н. Н.* Введение в теорию линейного и выпуклого программирования.— М. : Наука, 1976.— 192 с.
147. *Ермаков С. М.* Метод Монте-Карло и смежные вопросы.— М. : Наука, 1971.— 327 с.
148. *Ермольев Ю. М.* Методы решения нелинейных экстремальных задач.— Кибернетика, 1966, № 4, с. 1—17.
149. *Ермольев Ю. М.* О методе обобщенных стохастических градиентов и стохастических квазифейеровских последовательностях.— Кибернетика, 1969, № 2, с. 73—83.
150. *Ермольев Ю. М.* О некоторых проблемах стохастического программирования.— Кибернетика, 1970, № 1, с. 1—5.
151. *Ермольев Ю. М.* Об одной общей задаче стохастического программирования.— Кибернетика, 1971, № 3, с. 47—50.
152. *Ермольев Ю. М.* Стохастические модели и методы оптимизации.— Кибернетика, 1975, № 4, с. 109—119.
153. *Ермольев Ю. М.* Методы стохастического программирования.— М. : Наука, 1976.— 240 с.
154. *Ермольев Ю. М., Верченко П. И.* О методе линеаризации в предельных экстремальных задачах.— Кибернетика, 1976, № 2, с. 65—69.
155. *Ермольев Ю. М., Гупал А. М.* Аналог метода линеаризации в задачах минимизации недифференцируемых функций.— 1978, № 1, с. 65—68.
156. *Ермольев Ю. М., Некрылова З. В.* О некоторых методах стохастической оптимизации.— Кибернетика, 1966, № 6, с. 96—98.
157. *Ермольев Ю. М., Некрылова З. В.* Метод стохастических градиентов и его применение.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1967, № 1, с. 24—47.
158. *Ермольев Ю. М., Нурминский Е. А.* Экстремальные задачи статистики и численные методы стохастического программирования.— В кн.: Некоторые вопросы моделирования и управления систем. Киев : Наук. думка, 1973, с. 41—55.
159. *Ермольев Ю. М., Нурминский Е. А.* Предельные экстремальные задачи.— Кибернетика, 1973, № 4, с. 130—132.

160. *Ермольев Ю. М., Шор Н. З.* О минимизации недифференцируемых функций.— Кибернетика, 1967, № 1, с. 101—102.
161. *Жданов В. А.* Об одном методе покоординатного спуска.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 5, с. 1183—1188.
162. *Жилинскас А. Г.* Одношаговый байесовский алгоритм минимизации одномерных функций в присутствии помех.— В кн.: Теория оптимальных решений. Вильнюс: Изд. АН Лит. ССР, 1975, № 1, с. 9—22.
163. *Жилинскас А. Г.* Метод одномерной многоэкстремальной минимизации.— Изв. АН СССР. Техн. кибернетика, 1976, № 4, с. 71—74.
164. *Завриев С. К.* Комбинированный метод штрафов и стохастического градиента для поиска максимина.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 2, с. 329—343.
165. *Завриев С. К.* Об отыскании стационарных точек в задаче поиска максимина с ограничениями.— Вестник МГУ, Сер. «Вычисл. математика и кибернетика», 1980, № 2, с. 48—57.
166. *Зак Ю. А.* Принцип разложения и итеративные методы решения задач линейного программирования большого размера.— Автоматика и телемеханика, 1971, № 12, с. 143—150.
167. *Зангвил У. И.* Нелинейное программирование. Единый подход.— М. : Сов. радио, 1973.— 312 с.
168. *Звягина Р. А.* Задачи линейного программирования с блочно-диагональными матрицами.— В кн.: Оптимальное планирование. Новосибирск, 1964, вып. 2, с. 50—61.
169. *Звягина Р. А.* Задачи линейного программирования с матрицами произвольной блочной структуры. Докл. АН СССР, 1971, т. 196, № 4, с. 755—758.
170. *Звягина Р. А.* Об общем методе решения задач линейного программирования блочной структуры.— В кн.: Оптимизация. Новосибирск, 1971, № 1 (18), с. 22—40.
171. *Зинько П. Н.* О некоторых методах отыскания экстремальных траекторий систем дифференциальных уравнений: Автореф. дис. ... канд. физ.-мат. наук.— Киев, 1975.— 14 с.
172. *Зойтендейк Г.* Методы возможных направлений.— М.: Изд-во иностр. лит., 1963.— 175 с.
173. *Зуховицкий С. И., Авдеева Л. И.* Линейное и выпуклое программирование.— М. : Наука, 1967.— 460 с.
174. *Зуховицкий С. И., Поляк Р. А., Примак М. Е.* Алгоритм для решения задач выпуклого чебышевского приближения.— Докл. АН СССР, 1963, т. 151, № 1, с. 27—30.
175. *Зуховицкий С. И., Поляк Р. А., Примак М. Е.* Алгоритм для решения задачи выпуклого программирования.— Докл. АН СССР, 1963, т. 153, № 5, с. 991—994.
176. *Интрилигатор М.* Математические методы оптимизации и экономическая теория.— М. : Прогресс, 1975.— 608 с.
177. *Иванов В. В.* Вопросы точности и эффективности вычислительных алгоритмов.— Киев : Изд. ИК АН УССР, 1969.— 135 с.
178. *Иванов В. В.* Об оптимальных алгоритмах минимизации функций некоторых классов.— Кибернетика, 1972, № 4, с. 81—94.
179. *Иванов В. В., Людвиченко В. А.* Об одном методе последовательной безусловной минимизации решения задач математического программирования.— Кибернетика, 1977, № 2, с. 1—8.
180. *Иванов В. В., Трутень В. Е.* К методу штрафных функций.— Кибернетика, 1968, № 2, с. 34—39.
181. *Ильин В. А.* О работах А. Н. Тихонова по методам решения некорректно поставленных задач.— Успехи мат. наук, 1967, т. 22, вып. 2, с. 168—175.
182. *Иослович И. В., Макаренков Ю. М.* О методах сокращения размерности задач линейного программирования.— Экономика и мат. методы, 1975, т. 11, вып. 3, с. 516—524.
183. *Иоффе А. Д., Тихомиров В. М.* Теория экстремальных задач.— М. : Наука, 1974.— 479 с.

184. *Истомин Л. А.* Модификация метода Хоанг Туя минимизации вогнутой функции на многограннике.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 6, с. 1592—1597.
185. *Канторович Л. В.* О методе наискорейшего спуска.— Докл. АН СССР, 1947, т. 56, № 3, с. 233—236.
186. *Канторович Л. В.* О методе Ньютона.— Тр. Ин-та математики АН СССР, 1949, т. 28, с. 104—144.
187. *Канторович Л. В.* Экономический расчет наилучшего использования ресурсов.— М. : Изд-во АН СССР, 1959.— 344 с.
188. *Каплинский А. И., Пропой А. И.* О стохастическом подходе к задачам нелинейного программирования.— Автоматика и телемеханика, 1970, № 3, с. 122—133.
189. *Каплинский А. И., Позняк А. С., Пропой А. И.* Условия оптимальности для некоторых задач стохастического программирования.— Автоматика и телемеханика, 1971, № 8, с. 51—60.
190. *Карабегов В. И.* Об одной параметрической задаче линейного программирования.— Журн. вычисл. математики и мат. физики, 1963, т. 3, № 3, с. 547—558.
191. *Карлин С.* Математические методы в теории игр, программировании и экономике.— М. : Мир, 1964.— 840 с.
192. *Карманов В. Г.* Оценки сходимости итерационных методов минимизации.— Журн. вычисл. математики и мат. физики, 1974, т. 14, № 1, с. 3—14.
193. *Карманов В. Г.* Сходимость метода случайного поиска в задачах выпуклой минимизации.— Теория вероятности и ее применения, 1974, т. 19, № 4, с. 817—824.
194. *Карманов В. Г.* Математическое программирование.— М. : Наука, 1975.— 272 с.
195. *Катковник В. Я.* Метод операторов усреднения в итерационных алгоритмах решения стохастических экстремальных задач.— Кибернетика, 1972, № 4, с. 123—131.
196. *Катковник В. Я.* Линейные оценки и стохастические задачи оптимизации (метод параметрических операторов усреднения).— М. : Наука, 1976.— 488 с.
197. *Катковник В. Я., Кульчицкий О. Ю.* Сходимость одного алгоритма случайного поиска.— Автоматика и телемеханика, 1972, № 8, с. 81—87.
198. *Келли Генри Дж.* Метод градиентов.— В кн.: Методы оптимизации / Под ред. Дж. Лейтмана.— М. : Наука, 1965, с. 244—306.
199. *Ким К. В.* Об использовании специфики условий задачи в методе улучшения плана.— Экономика и мат. методы, 1965, т. 1, вып. 1, с. 86—93.
200. *Коллатц Л.* Функциональный анализ и вычислительная математика.— М. : Мир, 1969.— 447 с.
201. *Кононенко А. Ф.* О многошаговых конфликтах с обменом информации.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 4, с. 922—931.
202. *Коротченко А. Г.* Об одном алгоритме поиска наибольшего значения одномерных функций.— Журн. вычисл. математики и мат. физики, 1978, т. 18, № 3, с. 563—573.
203. *Красовский Н. Н., Субботин А. И.* Позиционные дифференциальные игры.— М. : Наука, 1974.— 456 с.
204. *Кругер А. Я.* Обобщенные производные по направлениям и необходимые условия экстремума в невыпуклых задачах математического программирования.— В кн.: Тез. докл. IV Всесоюз. конф. по пробл. теорет. кибернетики. Новосибирск, 1977, с. 60—62.
205. *Кудин И. Б.* О решении общей задачи линейного программирования.— Экономика и мат. методы, 1975, т. 11, № 4, с. 794—795.
206. *Кукушкин П. С.* Об одной игре с неполной информацией.— Журн. вычисл. математики и мат. физики, т. 13, № 1, 1973, с. 210—216.
207. *Кун Г.* Венгерский метод решения задачи о назначениях.— В кн.: Методы и алгоритмы решения транспортной задачи. М. : Госстатиздат, 1963, с. 35—52.
208. *Кудринский В. Ю., Трутень В. Е.* Решение системы линейных алгебраиче-

ских уравнений на ЕС ЭВМ с помощью пакета СПАУ.— Тез. докл. V Всесоюз. симпоз.— М. : Изд-во ЦЭМИ АН СССР, 1978, с. 174—175.
209. *Курдюмов И. В., Мосолова М. В., Назайкинский В. Е.* Вычислительный алгоритм для задачи квадратичного программирования большой размерности.— Журн. вычисл. математики и мат. физики, 1978, т. 18, № 5, с. 1119—1128.
210. *Куржанский А. Б.* Управление и наблюдение в условиях неопределенности.— М. : Наука, 1977.— 392 с.
211. *Кутиков Л. М.* Декомпозиция блочных задач линейного программирования со слабо связанными блоками.— Экономика и мат. методы, 1973, т. 9, вып. 4, с. 739—743.
212. *Кухтенко А. И.* Проблемы многомерности в теории сложных систем.— В кн.: Кибернетика и вычислительная техника. Сложные системы управления. Киев : Наук. думка, 1969, вып. 1, с. 6—35.
213. *Кушнер Г. Дж.* Стохастическая устойчивость и управление.— М. : Мир, 1969.— 200 с.
214. *Кюнци Г. П., Крелле В.* Нелинейное программирование.— М. : Сов. радио, 1965.— 301 с.
215. *Лебедев В. Ю.* Итерационный алгоритм решения задач линейного программирования.— Журн. вычисл. математики и мат. физики, 1973, т. 13, № 6, с. 1527—1533.
216. *Лебедев В. Ю.* Приближенный алгоритм решения задачи линейного программирования.— Журн. вычисл. математики и мат. физики, 1974, т. 14, № 4, с. 1052—1058.
217. *Лебедев В. Ю.* Схема поиска приближенных решений задачи выпуклого программирования.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 1, с. 249—253.
218. *Лебедев В. Ю.* О сходимости метода нагруженного функционала в задачах выпуклого программирования.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 3, с. 765—768.
219. *Левин А. Ю.* Об одном алгоритме минимизации выпуклых функций.— Докл. АН СССР, 1965, т. 160, № 6, с. 1244—1247.
220. *Левитин Е. С.* Об одном методе минимизации для негладких экстремальных задач.— Журн. вычисл. математики и мат. физики, 1969, т. 9, № 4, с. 783—807.
221. *Левитин Е. С.* Об оценках устойчивости по решению в задачах безусловной оптимизации.— Журн. вычисл. математики и мат. физики, 1976, т. 16, № 3, с. 585—596.
222. *Левитин Е. С., Поляк Б. Т.* Методы минимизации при наличии ограничений.— Журн. вычисл. математики и мат. физики, 1966, т. 6, № 5, с. 787—823.
223. *Лемке Дж.* Двойственный метод решения задач линейного программирования.— В кн.: Методы решения общей задачи линейного программирования.— М. : Госстатиздат, 1963, с. 55—70.
224. *Лепеха Н. П.* Об алгоритме случайного поиска в задачах идентификации.— В кн.: Вопросы оптимизации в динамических системах с непрерывно-дискретными параметрами. Киев : Наук. думка, 1980, с. 164—171.
225. *Любич Ю. И., Майстровский Г. Д.* Общая теория релаксационных процессов для выпуклых функционалов.— Успехи мат. науки, 1970, т. 25, вып. 1, с. 57—112.
226. *Ляшенко И. Н., Карагодова Е. А., Черникова Н. В., Шор Н. З.* Линейное и нелинейное программирование.— Киев : Вища школа, 1975.— 372 с.
227. *Ляшко И. И., Кудринский В. Ю., Остапчук В. С.* О решении плохо обусловленных систем линейных алгебраических уравнений с положительно-определенными матрицами.— Докл. АН СССР, 1979, т. 245, № 3, с. 533—536.
228. *Ляшко И. И., Макаров В. Л., Скоробогатько А. А.* Методы вычислений.— Киев : Вища школа, 1977.— 408 с.
229. *Майстровский Г. Д.* Доказательство квадратичной сходимости метода сопряженных градиентов.— В кн.: Вычисл. математика и вычисл. техника. Харьков : Изд. ФТИНТ АН СССР, 1971, вып. 2, с. 3—6.

230. *Майстровский Г. Д.* О сходимости метода сопряженных градиентов.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 5, с. 1291—1294.
231. *Майстровский Г. Д.* О градиентных методах отыскания седловых точек.— Экономика и мат. методы, 1976, т. 12, вып. 5, с. 917—929.
232. *Майстровский Г. Д.* Метод сопряженных градиентов в задаче условной минимизации.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 2, с. 408—501.
233. *Майстровский Г. Д.* Градиентный метод для модифицированной функции Лагранжа.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 1, с. 56—69.
234. *Майстровский Г. Д.* Локально квадратично-сходящийся метод условной минимизации, использующий модифицированный лагранжиан.— Журн. вычисл. математики и мат. физики, 1980, т. 20, № 1, с. 27—37.
235. *Майстровский Г. Д., Ольховский Ю. Г.* К теории условной минимизации.— Докл. АН СССР, 1973, т. 212, № 4, с. 818—821.
236. *Майстровский Г. Д., Ольховский Ю. Г.* О скорости сходимости метода наискорейшего спуска в задаче условной минимизации.— Журн. вычисл. математики и мат. физики, 1975, т. 15, № 4, с. 844—859.
237. *Малинников В. В.* Метод разложения в решении больших задач линейного программирования с блочной структурой.— Экономика и мат. методы, 1971, т. 7, вып. 5, с. 733—736.
238. *Малоземов В. Н.* О сходимости сеточного метода в задаче наилучшей полиномиальной аппроксимации.— Вестник ЛГУ, 1970, № 19, с. 138—140.
239. *Малоземов В. Н.* О сходимости сеточного метода в нелинейных минимаксных задачах.— Вестник ЛГУ, 1971, № 19, с. 35—37.
240. *Малоземов В. Н.* О выравнивании максимумов.— Журн. вычисл. математики и мат. физики, 1976, т. 16, № 3, с. 781—784.
241. *Манкрес Дж.* Алгоритм решения задачи выбора и транспортной задачи.— В кн.: Методы и алгоритмы решения транспортной задачи. М. : Госстатиздат, 1963, с. 73—79.
242. *Марчук Г. И.* Методы вычислительной математики.— Новосибирск : Наука, 1973,— 352 с.
243. *Маш В. А.* Об одном способе выбора вводимой переменной в симплексном методе линейного программирования.— Журн. вычисл. математики и мат. физики, 1964, т. 4, № 2, с. 376—379.
244. *Мелешко В. И.* Рекуррентное статистическое оценивание на основе псевдообратных операторов.— Автоматика и телемеханика, 1976, № 8, с. 101—110.
245. *Мелешко В. И.* Методы псевдообратных операторов определения седловых точек.— В кн.: Численные методы нелинейного программирования. Киев : Наук. думка, 1976, с. 27—32.
246. *Мелешко В. И.* Об устойчивом рекуррентном псевдообращении вырожденных и плохо обусловленных матриц.— В кн.: Численные методы нелинейного программирования. Киев : Наук. думка, 1976, с. 53—63.
247. *Мелешко В. И.* Численные методы безусловной оптимизации с использованием псевдообратных операторов.— В кн.: Численные методы нелинейного программирования: Тез. II Всесоюз. семинара, Харьков, 28 мая — 3 июня 1976. Харьков : Вища школа Изд-во при Харьк. ун-те, 1976, с. 14—22.
248. *Мелешко В. И.* Устойчивое к возмущениям псевдообращение замкнутых операторов.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 5, с. 1132—1141.
249. *Мелешко В. И.* Исследование алгоритмов с восстановлением гессиана по значениям градиентов.— Кибернетика, 1977, № 5, с. 91—99.
250. *Мелешко В. И.* Ускоренные алгоритмы адаптации в задачах с неопределенностью на основе теории псевдообратных операторов.— Изв. АН СССР. Техн. кибернетика, 1978, № 2, с. 39—52.
251. *Мелешко В. И.* Псевдообратные операторы и рекуррентное вычисление псевдорешений в гильбертовых пространствах.— Сибир. мат. журн, 1978, № 1, с. 108—121.
252. *Мелешко В. И., Песина Р. И., Башмакова Т. И.* Экспериментальное исследование регуляризованных рекуррентных псевдообращений матриц.—

В кн.: Численные методы нелинейного программирования. Киев : Наук. думка, 1976, с. 68—78.
253. *Метод* статистических испытаний (метод Монте — Карло) / Под ред. Ю. А. Шрейдера.— М. : Физматгиз, 1962.— 331 с.
254. *Митчелл Б. Ф., Демьянов В. Ф., Малоземов В. Н.* Нахождение ближайшей к началу координат точки многогранника.— Вестн. ЛГУ, 1971, № 13, с. 38—45.
255. *Михалевич В. С.* Последовательные алгоритмы оптимизации и их применение. Об одной схеме последовательного поиска.— Кибернетика, 1965, № 1, с. 45—46; № 2, с. 85—89.
256. *Михалевич В. С., Волкович В. Л.* О некоторых математических эвристических особенностях процесса проектирования сложных систем.— Управляющие системы и машины, 1976, № 3, с. 3—9.
257. *Михалевич В. С. и др.* Вычислительные методы выбора оптимальных проектных решений.— Киев : Наук. думка, 1977.— 178 с.
258. *Мовшович С. М.* Метод невязок для решения задач блочной структуры.— Экономика и мат. методы, 1966, т. 2, вып. 4, с. 571—577.
259. *Моисеев Н. Н.* Математические задачи системного анализа.— М. : Наука, 1981.— 488 с.
260. *Моисеев Н. Н.* Математика ставит эксперимент.— М. : Наука, 1979.— 224 с.
261. *Моисеев Н. Н., Иванилов Ю. П., Столярова Е. М.* Методы оптимизации.— М. : Наука, 1978.— 352 с.
262. *Моцкус Й. Б.* О байесовых методах поиска экстремума.— Автоматика и телемеханика, 1972, № 3, с. 53—62.
263. *Невельсон М. Б., Хасьминский Р. З.* Стохастическая аппроксимация и рекуррентное оценивание.— М. : Наука, 1972.— 304 с.
264. *Немировский А. С., Юдин Д. Б.* Методы оптимизации, адаптивные к «существенной» размерности задачи.— Автоматика и телемеханика, 1977, № 4, с. 75—87.
265. *Николаев Е. Г.* О скорейшем спуске, основанном на методе случайного G^m-градиента.— Автоматика и вычисл. техника, 1970, № 3, с. 40—46.
266. *Новикова Н. М.* Стохастический квазиградиентный метод поиска максимина.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 1, с. 91—99.
267. *Нурминский Е. А.* Условия сходимости алгоритмов нелинейного программирования.— Кибернетика, 1972, № 6, с. 79—81.
268. *Нурминский Е. А.* О свойствах одного класса функций.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1972, с. 92—96.
269. *Нурминский Е. А.* Квазиградиентный метод решения задачи нелинейного программирования.— Кибернетика, 1973, № 1, с. 122—125.
270. *Нурминский Е. А.* Сходимость стохастических аналогов детерминированных процедур нелинейного программирования.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1973, с. 60—73.
271. *Нурминский Е. А.* Условия сходимости алгоритмов стохастического программирования.— Кибернетика, 1973, № 3, с. 84—87.
272. *Нурминский Е. А.* Численные методы решения детерминированных и стохастических минимаксных задач.— Киев : Наук. думка, 1979.— 161 с.
273. *Нурминский Е. А., Верченко П. И.* О сходимости алгоритмов поиска седловых точек.— Кибернетика, 1977, № 3, с. 112—116.
274. *Нурминский Е. А., Желиховский А. А.* Исследование одной регулировки шага в квазиградиентном методе минимизации слабо выпуклых функций.— Кибернетика, 1974, № 6, с. 101—105.
275. *Нурминский Е. А., Желиховский А. А.* ε-квазиградиентный метод решения негладких экстремальных задач.— Кибернетика, 1977, № 1, с. 109—113.
276. *Обломская Л. Я.* Сравнение быстроты сходимости методов сопряженных градиентов и градиентного для квадратичных функционалов.— В кн.: Вопросы точности и эффективности вычислительных алгоритмов, 1968, № 4, с. 94—103.
277. *Овруцкий И. Г., Шор Н. З.* Применение методов минимизации негладких функций для решения задач интерпретации гравиметрических наблюдений.— Кибернетика, 1976, № 2, с. 57—64.

278. *Ортега Дж., Рейнболдт В.* Итерационные методы решения нелинейных систем уравнений со многими неизвестными.— М. : Мир, 1975.— 560 с.
279. *Панин В. М.* О методе Ньютона с регулировкой шага для задач дискретного минимакса.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1977, с. 46—55.
280. *Панин В. М.* О методе второго порядка для задачи дискретного минимакса.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 1, с. 88—98.
281. *Пиявский С. А.* Алгоритм отыскания абсолютного минимума функций.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1967, № 2, с. 13—24.
282. *Пиявский С. А.* Один алгоритм отыскания абсолютного экстремума функции.— Журн. вычисл. математики и мат. физики, 1972, т. 12, № 4, с. 888—896.
283. *Подвальный Л. Д.* О сходимости метода сопряженных случайных направлений.— Кибернетика, 1977, № 1, с. 133—137.
284. *Подвальный Л. Д., Пугачева Л. Н.* Метод сопряженных случайных направлений в задачах овражного типа.— Кибернетика, 1976, № 4, с. 49—51.
285. *Полак Э.* Численные методы оптимизации. Единый подход.— М. : Мир, 1974,— 376 с.
286. *Поляк Б. Т.* Градиентные методы минимизации функционалов.— Журн. вычисл. математики и мат. физики, 1963, т. 3, № 4, с. 643—654.
287. *Поляк Б. Т.* О некоторых способах ускорения сходимости итерационных методов.— Журн. вычисл. математики и мат. физики, 1964, т. 4, № 5, с. 791—803.
288. *Поляк Б. Т.* Один общий метод решения экстремальных задач.— Докл. АН СССР, 1967, т. 174, № 1, с. 33—36.
289. *Поляк Б. Т.* Методы минимизации функций многих переменных.— Экономика и мат. методы, 1967, т. 3, вып. 6, с. 881—908.
290. *Поляк Б. Т.* Минимизация негладких функционалов.— Журн. вычисл. математики и мат. физики, 1969, т. 9, № 3, с. 509—521.
291. *Поляк Б. Т.* Метод сопряженных градиентов в задачах на экстремум.— Журн. вычисл. математики и мат. физики, 1969, т. 9, № 4, с. 807—821.
292. *Поляк Б. Т.* Метод сопряженных градиентов.— В кн.: Тр. II Зимней школы по мат. программированию и смежным вопросам. М. : 1969, вып. 1, с. 152—201.
293. *Поляк Б. Т.* Итерационные методы, использующие множители Лагранжа для решения экстремальных задач с ограничениями типа равенств.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 5, с. 1098—1106.
294. *Поляк Б. Т.* О скорости сходимости метода штрафных функций.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 1, с. 3—11.
295. *Поляк Б. Т.* Сходимость методов возможных направлений в экстремальных задачах.— Журн. вычисл. математики и мат. физики, 1971, т. 11, № 4, с. 855—869.
296. *Поляк Б. Т.* Сходимость и скорость сходимости итеративных стохастических алгоритмов. I. Общий случай.— Автоматика и телемеханика, 1976, № 12, с. 83—94.
297. *Поляк Б. Т.* Сходимость и скорость сходимости итеративных стохастических алгоритмов. II. Линейный случай.— Автоматика и телемеханика, 1977, № 4, с. 101—107.
298. *Поляк Б. Т.* Сравнение скорости сходимости одношаговых и многошаговых алгоритмов оптимизации при наличии помех.— Изв. АН СССР. Техн. кибернетика, 1977, № 1, с. 9—12.
299. *Поляк Б. Т.* Минимизация сложных функций регрессии.— Кибернетика, 1978, № 4, с. 148—149.
300. *Поляк Б. Т.* Методы решения задач на условный экстремум при наличии случайных помех.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 1, с. 70—78.
301. *Поляк Б. Т., Третьяков Н. В.* Об одном итерационном методе линейного программирования и его экономической интерпретации.— Экономика и мат. методы, 1972, т. 8, вып. 5, с. 740—751.
302. *Поляк Б. Т., Третьяков Н. В.* Метод штрафных оценок для задач на услов-

ный экстремум.— Журн. вычисл. математики и мат. физики, 1973, т. 13, № 1, с. 34—46.
303. *Поляк Б. Т., Цыпкин Я. З.* Псевдоградиентные алгоритмы адаптации и обучения.— Автоматика и телемеханика, 1973, № 3, с. 45—68.
304. *Поляк Р. А.* К ускорению сходимости методов выпуклого программирования.— Докл. АН СССР, 1973, т. 212, № 5, с. 1063—1066.
305. *Поляк Р. А.* Об одном общем принципе ускорения сходимости методов выпуклого программирования.— В кн.: Тр. Зимней школы по мат. программированию и смежных вопросах. М. : Изд-во ЦЭМИ, 1973, с. 113—153.
306. *Поляк Р. А.* Методы управляющих последовательностей в экстремальных задачах при наличии ограничений.— Журн. вычисл. математики и мат. физики, 1978, т. 18, № 1, с. 17—34.
307. *Примак М. Е.* Об одном вычислительном процессе отыскания точек равновесия.— Кибернетика, 1973, № 1, с. 91—96.
308. *Примак М. Е.* О сходимости метода чебышевских центров и метода центрированных сечений для решения задачи выпуклого программирования.— Докл. АН СССР, 1975, т. 222, № 2, с. 273—276.
309. *Примак М. Е.* О сходимости модифицированного метода чебышевских центров решения задачи выпуклого программирования.— Кибернетика, 1977, № 5, с. 100—102.
310. *Пропой А. И., Ядыкин А. Б.* Параметрическое квадратичное и линейное программирование. I.— Автоматика и телемеханика, 1978, № 2, с. 102—112.
311. *Пропой А. И., Ядыкин А. Б.* Параметрическое квадратичное и линейное программирование. II.— Автоматика и телемеханика, 1978, № 4, с. 135—143.
312. *Пшеничный Б. Н.* Выпуклое программирование в нормированном пространстве.— Кибернетика, 1965, № 5, с. 46—54.
313. *Пшеничный Б. Н.* Об одном алгоритме спуска.— Журн. вычисл. математики и мат. физики, 1968, т. 8, № 3, с. 649—652.
314. *Пшеничный Б. Н.* Необходимое условие экстремума.— М. : Наука, 1969.— 152 с.
315. *Пшеничный Б. Н.* Алгоритм для общей задачи математического программирования.— Кибернетика, 1970, № 5, с. 120—125.
316. *Пшеничный Б. Н.* Метод минимизации функций без вычисления производных.— Кибернетика, 1973, № 4, с. 127—129.
317. *Пшеничный Б. Н.* Метод минимизации функций без вычисления производных.— Докл. АН СССР, 1977, т. 235, № 5, с. 1026—1029.
318. *Пшеничный Б. Н.* О необходимых условиях экстремума для негладких функций.— Кибернетика, 1977, № 6, с. 92—96.
319. *Пшеничный Б. Н., Ганжела И. Ф.* Алгоритм для решения задачи выпуклого программирования при линейных ограничениях.— Кибернетика, 1970, № 3, с. 81—85.
320. *Пшеничный Б. Н., Данилин Ю. М.* Численные методы в экстремальных задачах.— М. : Наука, 1975.— 320 с.
321. *Пшеничный Б. Н., Марченко Д. И.* Об одном подходе к нахождению глобального минимума.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1967, № 2, с. 3—13.
322. *Пшеничный Б. Н., Редковский Н. Н.* Об одном численном методе минимизации без вычисления производных.— Журн. вычисл. математики и мат. физики, 1976, т. 16, с. 1388—1396.
323. *Пшеничный Б. Н., Редковский Н. Н.* О методе минимизации вдоль собственных векторов матрицы, близкой к матрице Гессе.— Кибернетика, 1977, № 5, с. 68—74.
324. *Пшеничный Б. Н., Редковский Н. Н.* Некоторые методы безусловной минимизации.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 5, с. 1127—1133.
325. *Разинкова Т. Л.* Об одном алгоритме машинного поиска экстремума функции.— Журн. вычисл. математики и мат. физики, 1965, т. 5, № 4, с. 737—742.
326. *Растригин Л. А.* Статистические методы поиска.— М. : Наука, 1968.— 376 с.

327. *Растригин Л. А.* Случайный поиск в процессах адаптации.— Рига : Зинатне, 1973.— 130 с.
328. *Растригин Л. А.* Системы экстремального управления.— М. : Наука, 1974.— 632 с.
329. *Растригин Л. А., Тарасенко Г. С.* Об одном адаптивном алгоритме случайного поиска.— В кн.: Проблемы случайного поиска. Рига : Зинатне, 1974, вып. 3, с. 108—112.
330. *Рвачев В. Л., Ганжела Н. Ф.* Решение нелинейных краевых задач методами математического программирования.— Вестн. Харьков. политехн. ин-та, 1975, № 105, с. 16—20.
331. *Редковский Н. Н.* Использование методов сопряженных направлений для минимизации без вычисления производных.— В кн.: Теория оптимальных решений. Киев : Изд ИК АН УССР, 1976.
332. *Ремез Е. Я.* Основы численных методов чебышевского приближения.— Киев : Наук. думка, 1969.— 623 с.
333. *Робинсон Дж.* Итеративный метод решения игр.— В кн.: Матричные игры. М. : Физматгиз, 1961, с. 110—117.
334. *Рокаффеллар Р.* Выпуклый анализ.— М. : Мир, 1973.— 472 с.
335 *Романовский И. В.* Алгоритмы решения экстремальных задач.— М. : Наука, 1977.— 351 с.
336. *Самарский А. А., Николаев Е. С.* Методы решения сеточных уравнений.— М. : Наука, 1978.— 290 с.
337. *Самойленко Ю. И.* Приведение к элементарной ячейке линейных систем с дискретной группой симметрии.— В кн.: Кибернетика и вычисл. техника, 1970, вып. 6, с. 21—35.
338. *Сеа Ж.* Оптимизация. Теория и алгоритмы.— М. : Мир, 1973.— 244 с.
339. *Сергиенко И. В.* О применении вектора спада для решения задач оптимизации комбинаторного типа.— Управляющие системы и машины, 1975, № 2, с. 86—94.
340 *Скоков В. А.* Замечание к методам минимизации, использующим операцию растяжения пространства.— Кибернетика, 1974, № 4, с. 115—117.
341. *Случайный* поиск (теория и применение). Систематический указатель литературы / Под ред. Л. А. Растригина.— Рига : Зинатне, 1972.— 78 с.
342. *Сотсков А. К.* Необходимые условия минимума для одного типа негладких задач.— Докл. АН СССР, 1969, т. 189, № 2, с. 261—269.
343. *Стронгин Р. Г.* Алгоритмы для поиска абсолютного минимума.— В кн.: Задачи статистической оптимизации / Под ред. Растригина Л. А.— Рига : Зинатне, 1971, с. 51—68.
344. *Стронгин Р. Г.* Рандомизация стратегий в поиске глобального экстремума.— В кн.: Проблемы случайного поиска, Рига : Зинатне, 1973, № 2, с. 19—30.
345. *Стронгин Р. Г.* О сходимости одного алгоритма поиска глобального экстремума.— Изв. АН СССР. Техн. кибернетика, 1973, № 4, с. 10—16.
346. *Стронгин Р. Г.* Численные методы в многоэкстремальных задачах (Информационно-статистические алгоритмы).— М. : Наука, 1978.— 240 с.
347. *Сыров Ю. П., Чурквеидзе Ш. С.* Вопросы оптимизации межотраслевых и межрайонных связей при планировании развития единой народнохозяйственной системы.— Иркутск : Изд. Иркут. ин-та народн. хоз-ва, 1970.— 78 с.
348. *Табак Д., Куо Б.* Оптимальное управление и математическое программирование.— М. : Наука, 1975.— 280 с.
349. *Тарасенко Г. С.* Сходимость адаптивного алгоритма случайного поиска.— Кибернетика, 1977, № 5, с. 88—90.
350. *Тарасова В. П.* Оптимальные стратегии поиска приближенного глобального экстремума для некоторого класса функций.— В кн.: Вычислительные системы. Новосибирск : СО АН СССР, 1976, вып. 67, с. 77—86.
351. *Тарасова В. П.* Оптимальные стратегии поиска области наибольших значений для некоторого класса функций.— Журн. вычисл. математики и мат. физики, 1978, т. 18, № 4, с. 886—896.
352. *Телле В.* Применение модифицированной функции Лагранжа в блочном программировании.— Экономика и мат. методы, 1975, № 3, с. 525—534.

353. *Тимохин С. Г.* Декомпозиционный подход к решению задачи линейного программирования.— Экономика и мат. методы, 1977, № 2, с. 330—341.
354. *Тихонов А. Н.* Об устойчивости задач оптимизации функционалов.— Журн. вычисл. математики и мат. физики, 1966, т. 6, № 4, с. 631—634.
355. *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач.— М. : Наука, 1974.— 223 с.
356. *Тихонов А. Н., Гласко В. Б.* Применение метода регуляризации в нелинейных задачах.— Журн. вычисл. математики и мат. физики, 1965, т. 5, № 3, с. 463—473.
357. *Третьяков Н. В.* Метод штрафных оценок для задач выпуклого программирования.— Экономика и мат. методы, 1973, № 3, с. 526—540.
358. *Уайлд Д.* Методы поиска экстремума.— М. : Наука, 1967.— 267 с.
359. *Удзава X.* Итерационные методы вогнутого программирования.— В кн.: Исследования по линейному и нелинейному программированию. М. : Изд-во иностр. лит., 1962, с. 228—245.
360. *Уилкинсон Дж. X., Райнш К.* Справочник алгоритмов на языке Алгол. Линейная алгебра.— М. : Машиностроение, 1976.— 390 с.
361. *Ульм С. Ю.* Метод декомпозиции для решения задач математического программирования.— Изв. АН ЭССР. Физика, математика, 1973, т. 22, № 1, с. 93—95.
362. *Фадеев Д. К., Фадеева В. Н.* Вычислительные методы линейной алгебры.— М. : Физматгиз, 1960.— 656 с.
363. *Федоренко Р. П.* Опыт итерационного решения задач линейного программирования.— Журн. вычисл. математики и мат. физики, 1965, т. 5, № 4. с. 709—717.
364. *Федоренко Р. П.* Приближенное решение задач линейного программирования высокой размерности.— В кн.: Вычислительные методы линейной алгебры. Новосибирск : СО АН СССР, 1969, с. 74—86.
365. *Федоренко Р. П.* Итерационное решение задач линейного программирования.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 4, с. 895—907.
366. *Федоренко Р. П.* Об итерационном решении задач линейного программирования.— Журн. вычисл. математики и мат. физики, 1972, т. 12, № 2, с. 298—308.
367. *Федоренко Р. П.* О сходимости одного итерационного метода решения задач линейного программирования.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 6, с. 1369—1407.
368. *Федоров В. В.* О методе штрафных функций в задаче определения максимина.— Журн. вычисл. математики и мат. физики, 1972, т. 12, № 2, с. 321—333.
369. *Федоров В. В.* К задаче об искусственном рассеивании.— В кн.: Исследование операций. М. : Изд. ВЦ АН СССР, 1974, вып. 4, с. 200—218.
370. *Федоров В. В.* Методы поиска максимина.— М. : Изд-во МГУ, 1975, вып. 1.— 88 с.
371. *Федоров В. В.* Методы поиска максимина.— М. : Изд-во МГУ, 1976, вып. 2.— 104 с.
372. *Федоров В. В.* Численные методы максимина.— М. : Наука, 1979.— 280 с.
373. *Фиакко А., Мак-Кормик Г. П.* Нелинейное программирование. Методы последовательной безусловной минимизации.— М. : Мир, 1972.— 240 с.
374. *Форд Л., Фулдерсон Д.* Решение транспортной задачи.— В кн.: Методы и алгоритмы решения транспортной задачи. М. : Госстатиздат, 1963, с. 61—72.
375. *Фридман В. М.* О сходимости методов типа наискорейшего спуска.— Успехи мат. наук, 1962, т. 17, № 3, с. 201—204.
376. *Хедли Дж.* Нелинейное и динамическое программирование.— М. : Мир, 1967.— 507 с.
377. *Хенкин М. З.* О выборе штрафных коэффициентов, сохраняющем скорость сходимости метода Ньютона.— Кибернетика, 1979, № 6, с. 46—50.
378. *Химмельблау Д.* Прикладное нелинейное программирование.— М. : Мир, 1975.— 536 с.
379. *Цыпкин Я. З.* Адаптация и обучение в автоматических системах.— М. : Наука, 1968.— 400 с.

380. *Цыпкин Я. З.* Обобщенные алгоритмы обучения.— Автоматика и телемеханика, 1970, № 1, с. 97—103.
381. *Чан Хань.* Приближенные методы решения задач выпуклого программирования.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 4, с. 805—815.
382. *Чан Хань.* Некоторые методы минимизации при наличии ограничений в гильбертовом пространстве.— Журн. вычисл. математики и мат. физики, 1977, т. 17, № 6, с. 1387—1395.
383. *Чан Хань.* Некоторые новые методы решения конечномерных экстремальных задач : Автореф. дис. ... канд. физ.-мат. наук.— М. : 1978.— 18 с.
384. *Чан Хань.* Некоторые методы минимизации без ограничений.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 2, с. 304—315.
385. *Черноусько Ф. Л.* Метод локальных вариаций для численного решения вариационных задач.— Журн. вычисл. математики и мат. физики, 1965, т. 5, № 4, с. 749—754.
386. *Черноусько Ф. Л.* Об оптимальном поиске экстремума унимодальных функций.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 4, с. 922—933.
387. *Черноусько Ф. Л.* Об оптимальном поиске минимума выпуклых функций.— Журн. вычисл. математики и мат. физики, 1970, т. 10, № 6, с. 1355—1366.
388. *Черноусько Ф. Л., Баничук Н. В.* Вариационные задачи механики и управления.— М. : Наука, 1973.— 238 с.
389. *Черноусько Ф. Л., Меликян А. А.* Игровые задачи управления и поиска.— М. : Наука, 1972.— 271 с.
390. *Шаманский В. Е.* Методы численного решения краевых задач на ЭЦВМ. В 2-х ч.— Киев : Изд-во АН УССР, 1963.— 200 с.
391. *Шепилов М. А.* О градиентных и штрафных методах в задачах математического программирования: Автореф. дис. ... канд. физ.-мат. наук.— М., 1974.— 18 с.
392. *Шепилов М. А.* О методе обобщенного градиента отыскания абсолютного минимума выпуклой функции.— Кибернетика, 1976, № 4, с. 52—57.
393. *Шепилов М. А.* О методе штрафов для задач выпуклого программирования.— Журн. вычисл. математики и мат. физики, 1979, т. 19, № 4, с. 889—895.
394. *Шор Н. З.* Применение обобщенного градиентного спуска в блочном программировании.— Кибернетика, 1967, № 3, с. 53—55.
395. *Шор Н. З.* О скорости сходимости обобщенного градиентного спуска.— Кибернетика, 1968, № 3, с. 98—99.
396. *Шор Н. З.* Обобщенный градиентный спуск.— В кн.: Тр. I Зимней школы по мат. программированию. М. : Изд. АН СССР, 1969, вып. 3, с. 578—585.
397. *Шор Н. З.* Методы минимизации недифференцируемых функций и их приложения : Автореф. дис. ... д-ра физ.-мат. наук.— Киев, 1970.— 44 с.
398. *Шор Н. З.* О скорости сходимости метода обобщенного градиентного спуска с растяжением пространства.— Кибернетика, 1970, № 2, с. 80—85.
399. *Шор Н. З.* О классе почти дифференцируемых функций и одном методе минимизации функций этого класса.— Кибернетика, 1972, № 4, с. 65—70.
400. *Шор Н. З.* Исследование сходимости метода градиентного типа с растяжением пространства в направлении разности двух последовательных градиентов.— Кибернетика, 1975, № 4, с. 48—53.
401. *Шор Н. З.* Обобщенные градиентные методы минимизации негладких функций и их применение к задачам математического программирования.— Экономика и мат. методы, 1976, вып. 2, с. 337—356.
402. *Шор Н. З.* Метод отсечения с растяжением пространства для решения задач выпуклого программирования.— Кибернетика, 1977, № 1, с. 94—95.
403. *Шор Н. З.* Новые направления в развитии методов негладкой оптимизации.— Кибернетика, 1977, № 6, с. 87—92.
404. *Шор Н. З.* Методы минимизации недифференцируемых функций и их приложения.— Киев : Наук. думка, 1979.— 200 с.
405. *Шор Н. З., Билецкий В. И.* Метод растяжения пространства для ускорения сходимости в задачах овражного типа.— В кн.: Теория оптимальных решений. Киев : Изд. ИК АН УССР, 1969, № 2, с. 3—18.

406. *Шор Н. З., Гамбург П. Р.* Некоторые вопросы сходимости обобщенного градиентного спуска.— Кибернетика, 1971, № 6, с. 82—84.
407. *Шор Н. З., Гершович В. И.* Об одном семействе алгоритмов для решения задач выпуклого программирования.— Кибернетика, 1979, № 4, с. 62—67.
408. *Шор Н. З., Журбенко Н. Г.* Метод минимизации, использующий операцию растяжения пространства в направлении разности двух последовательных градиентов.— Кибернетика, 1971, № 3, с. 51—59.
409. *Шор Н. З., Шабашова Л. П.* О решении минимаксных задач методом обобщенного градиентного спуска с растяжением пространства.— Кибернетика, 1972, № 1, с. 82—88.
410. *Энеев Т. М.* О применении градиентного метода в задачах теории оптимального управления.— Космические исследования, № 5, 1966, с. 72—80.
411. *Эрроу К. Дж., Гурвиц Л., Удзава Х.* Исследования по линейному и нелинейному программированию.— М. : Изд-во иностр. лит., 1962.— 334 с.
412. *Юдин Д. Б.* Математические методы управления в условиях неполной информации.— М. : Сов. радио, 1974.— 400 с.
413. *Юдин Д. Б.* Задачи и методы стохастического программирования.— М. : Сов. радио, 1979.— 392 с.
414. *Юдин Д. Б., Гольштейн Е. Г.* Линейное программирование.— М. : Физматгиз, 1963.— 776 с.
415. *Юдин Д. Б., Гольштейн Е. Г.* Задачи и методы линейного программирования.— М. : Сов. радио, 1964.— 736 с.
416. *Юдин Д. Б., Гольштейн Е. Г.* Линейное программирование. Теория, методы и приложения.— М. : Наука, 1969.— 424 с.
417. *Юдин Д. Б., Немировский А. С.* Информационная сложность и эффективные методы решения выпуклых экстремальных задач.— Экономика и мат. методы, 1976, т. 12, вып. 2, с. 357—369.
418. *Яблонский А. И.* Об экстремальных свойствах случайного поиска.— Изв. вузов. Радиофизика, 1971, т. 14, № 7, с. 1104—1111.
419. *Яковлев М. Н.* О некоторых методах решения нелинейных уравнений.— Тр. ин-та математики АН СССР, 1965, т. 84, с. 8—40.
420. *Abadie J., Guigou J.* Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints.— In: Optimization / Ed. R. Fletcher. London ; New York : Academic Press, 1969, p. 37—49.
421. *Altman M.* Generalized gradient methods of minimizing a functional.— Bull. Acad. Polon. Sci., **14**, № 6, 1966, p. 313—318.
422. *Armijo L.* Minimization of functions having continuous partial darivatives.— Pacific J. Math., № 16, 1966, p. 1—3.
423. *Antosiewicz H. A.* Newton's method and boundary value problem.— J. Comput. Sust. Sci., 1968, **2**, № 2, p. 177—202.
424. *Auslender A. and Brodeau F.* Convergence d'un algorithme de Frank et Wolfe appliqué à un problème de contrôle.— Rev. Française Informat. Rech. Opérationnelle, 1968, № 7, p. 3—12.
425. *Avriel Mordecal* Nonlinear programming. Analysis and Methods.— Prentice — Hall, INC., Englewood Cliffs, New Jersey, 1976, p. 112.
426. *Bard Y., Greenstadt J. L.* A modified Newton method for optimization with equality constraints.— In: Optimization/Ed. R. Fletcher. London; New York : Academic Press, 1969.
427. *Bazarra M. S.* An efficient cyclic coordinate method for optimizing penalty functions.— Nav. Res. Logist Quart., 1975, **22**, № 2, p. 399—404.
428. *Bazarra M. S., Good I. I., Shety C. M.* Optimality criteria in nonlinear programming without differentiability.— Oper. Res., 1971, **19**, № 1, p. 77—86.
429. *Beale E. M. L.* Cycling in the dual simplex algorithm.— Nav. Res. Logist. Quart., 1955, № 2, p. 269—275.
430. *Bertsecas D. P.* Necessary and sufficient conditions for a penalty method to be exact.— Math. Program., 1975, **9**, № 1, p. 87—89.
431. *Bertsecas D. P.* Combined primaldual and penalty methods for constrained minimization.— SIAM J. Contr. and Opt., 1975, **13**, № 3, p. 521—544.
432. *Bertsecas D. P.* On penalty and multiplier methods for constrained minimization.— SIAM J. Contr. and Opt., 1976, **14**, № 2, p. 216—235.

433. *Bertsecas D. P., Mitter K.* A descent numerical method for optimization problems with nondifferentiable cost functionals.— SIAM J Control, 1973, **11**, № 4, p. 637—652.
434. *Best Michael J., Ritter Klaus.* An accelerated conjugate direction method to solve linearly constrained minimization problems.— J. Compt. and Syst. Sci., 1975, **11**, № 3, p. 295—322.
435. *Blankenship J. W., Falk J. E.* Infinitely constrained optimization problems.— J. of Optim. Theory and Appl., 1976, **19**, № 2, p. 261—281.
436. *Blum J. R.* Multidimentional stochastic approximation methods.— Annals Math. Stat., 1954, **25**, № 4, p. 737—744.
437. *Boot J. C. G.* Quadratic Programming.— Amsterdam : North — Holland Publ. Co, 1964, p. 117.
438. *Brent R. P.* Algorithms for finding zeros and extrema of functions without calculating derivatives.— STAN—CS, February, 1971, p. 71—198.
439. *Brent R. P.* Algorithms for minimization without derivatives.— Englewood Cliffs; New Jersey : Prentice — Hall, 1973, p. 157.
440. *Broyden C. G.* Quasi — Newton methods.— In: Numerical Methods for Unconstrained Optimization / Ed. W. Murray. London; New — York : Akademic Press, 1972, p. 87—106.
441. *Brown G. W.* Iterative solutions of games by fuctitions play.— In: Activity Analysis of Production and Allocation / Ed. by T. C. Koopmans, Cowles Commission for Research in Economics Monograph. New — York : Wiley, 1951, № 13, p. 374—376.
442. *Bui — Trong — Lieu, Huard P.* La méthode des centres dans un espace topologique.— Num. Math., 1966, № 8, p. 65—67.
443. *Cabot A. Victor* Variations on a cutting plane method for solving concave minimization problems with linear constraints.— Nav. Res. Logist Quart., 1974, **21**, № 2, p. 265—274.
444. *Cannon M. D., Cullum C. D.* A tight upper bound on the rate of convergense of the Frank — Wolfe algorithm.— SIAM J. on control, 1968, **6**, p. 509—516.
445. *Cannon M. D., Cullum C. D., Polak E.* Theory of optimal control and mathematical programming.— New York, 1970, p. 131.
446. *Cauchy A. L.* Méthode générale pour la résolution des systémes d'équations simultanées, Comp. Rend. Acad. Sci., 1847, **25**, p. 536—538.
447. *Chaney R. W.* On the rate of convergence of some feasible direction algorithms.— J. of Optim. Theory and Appl. 1976, **20**, № 3, p. 297—313.
448. *Charalambous Ch.* Nonlinear least p-th optimization and nonlinear programming.— Math. Program., 1977, **12**, p. 195—225.
449. *Cheruault Y.* Une methode directe de minimization et applications.— Rev. Française Informat. Rech. Operationnelle, 1968, № 10, p. 31—52.
450. *Clarke F. H.* Generalized Gradients and Applications.— Transactions of the American Mathem. Society, 1975, **205**, № 3, p. 247—262.
451. *Clarke F. H.* A new approach to lagrang multipliers.— Math. Operations Res., 1976, № 1, p. 165—174.
452. *Conn A. R.* Constrained optimization using a non differentiable penalty function.— SIAM J. Num. Anal., 1973, **10**, p. 760—784.
453. *Curry H.* The method of steepest descent for nonlinear minimization problems.— Quart. Appl. Math., 1944, **2**, p. 258—261.
454. *Daniel James W.* Convergence of the conjugate gradient method with computationally convinient modifications.— Num. Math., 1967, **10**, № 2, p. 125—131.
455. *Daniel James W.* The Conjugate gradient method for linear and nonlinear operator equations.— SIAM J. Num. Anal., 1967, **4**, p. 10—26.
456. *Dantzig G. B.* A proof of the equivalence of the programming at the game problem.— In: Activity Analisys of Production and Alloçation / Ed. by Koopmans T. C., Cowles Commission Monograth, № 13, 1951, New York : Wiley, p. 330—335.
457. *Dantzig G. B., Wolfe Ph.* Decomposition algorithm for linear programs.— Econometrica, 1961, **29**, № 4, p. 767—778.
458. *Dantzig G. B., Wolfe Ph.* Decomposition prinsiple for linear programs.— Oper. Res., 1960, **8**, № 1, p. 101—111.

459. *Dunskin I.* The theory of max — min with application. SIAM J. Appl. Math., 1966, **14**, p. 641—664.
460. *Dutta S. R. K.* New algorithms for constrained minimax optimization. Math. Program., 1977, **13**, № 2, p. 140—155.
461. *Ecker J. G., Niemi F. D.* A dual method for quadratic programs with quadratic constraints. SIAM J. Appl. Math., 1975, **28**, № 3, p. 568—576.
462. *Elsinga J., Moore T.* A central cutting plane algorithm for the convex programming problem. Math. Program., 1975, **8**, № 1, p. 134—145.
463. *Fiacco A. V.* Penalty method for mathematical programming in with general constraint sets. J. of Optim. Theory and Appl., 1970, **6**, № 3, p. 252—268.
464. *Fiacco A. V., McCormick G. P.* The sequential unconstrained minimization technique for nonlinear programming, a primal — dual method. Management Sci., 1964, **10**, № 2, p. 360—366.
465. *Fletcher R.* Function minimization without evaluating derivatives. A review. Comput. J., 1965, **8**, № 1, p. 33—41.
466. *Fletcher R.* A new approach to variable metric algorithms. Comput. J., 1970, **13**, № 3, p. 317—322.
467. *Fletcher R.* A general quadratic programming algorithm. J. Inst. Maths. Applics., 1971, **7**, p. 76—91.
468. *Fletcher R.* An exact penalty function for nonlinear programming with inequalities. Math. Program., 1973, **5**, p. 129—150.
469. *Fletcher R., Freeman T. L.* A modified Newton method for minimization. J. of Optim. Theory and Appl., 1977, **23**, № 3, p. 357—372.
470. *Fletcher R., Powell M. J. D.* A rapidly convergent descent method for minimization. Comput. J., 1963, **6**, № 2, p. 163—168.
471. *Fletcher R., Reeves C. M.* Function minimization by conjugate gradients. Comput. J., 1964, **7**, № 2, p. 149—154.
472. *Fox R. L., Lasdon L. S., Tamir Arie, Ratner Margery*. An efficient one dimentional search procedure. Management Sci., 1975, **22**, № 1, p. 42—50.
473. *Frank M., Wolfe P.* An algorithm for quadratic programming. Nav. Res. Logist. Quart., 1956, **3**, № 1, p. 95—110.
474. *Daniel G., Luenberger D. G.* Efficiently converging minimization methods based on the reduced gradient. SIAM J. Contr. and Opt., 1976, **14**, № 1, p. 42—61.
475. *Gal T.* Rim multiparametric linear programming. Management Sci., 1975, **21**, № 5, p. 567—575.
476. *Garcia Palomares U. M., Mangasarian O. L.* Superlinearly convergent quasi-Newton algorithms for nonlinearly constrainted optimization problems. Math. program., 1976, **11**, № 1, p. 1—13.
477. *Gass S. I., Saaty T. L.* Parametric Objective Function, Part II, Generalization. J. of the Oper. Res. Society of America, 1955, № 3, p. 3—9.
478. *Gass S. I., Saaty T. L.* The computational algorithm for the parametric objective function. Nav. Res. Logist. Quart., 1955, № 2.
479. *Gill P. E., Murray W.* Quasi — Newton methods for unconstrained optimization. J. Inst. Maths. Applics., 1972, **9**, p. 91—108.
480. *Goffin I. L.* Nondifferentiable optimization. In: Abstracts IX Int. Symp. Math. Program. (Budapest-Hungary, aug. 23—27, 1976). Budapest: Bolyi Ianos Math. Soc., 1976.— 95 p.
481. *Goffin I. L.* On convergence rates of subgradient optimization. Math. Program., 1977, **13**, № 3, p. 329—347.
482. *Goldfarb D. A.* A family of variable metric methods derived by variational means. Math. Comput., 1970, **24**, № 109, p. 23—26.
483. *Goldstein A. A.* Optimization of Lipchitz continuous functions. Math. Program., 1977, 13, № 1, p. 14—22.
484. *Greenstadt J.* Variations on variable metric methods. Math. Comput., 1970, **24**, № 109, p. 1—22.
485. *Greenstadt J.* A quasi Newton method with no derivatives. Math. Comput., 1972, **26**, № 117, p. 145—166.
486. *Goldstein A. A., Price J. F.* An effective algorithm for minimization. Num. Math., 1967, **10**, p. 184—189.

487. *Haarhoff P. C., Buys J. D.* A new method for the optimization of nonlinear function subject to nonlinear constrained. Comput. J., 1970, **13**, № 2, p. 171—177.
488. *Han Shin Ping.* Superlinearly convergent variable metric algorithms for general nonlinear programming problems. Math. Programming, 1976, **11**, p. 263—282.
489. *Han Shin Ping.* A globally convergent method for nonlinear programming. J. of Optim. Theory and Appl., 1977, **22**, № 3, p. 297—309.
490. *Han Shin Ping.* Dual variable metric algorithms for constrained optimization. SIAM J. Contr. and Opt., 1977, **15**, № 4, p. 546—565.
491. *Held M., Wolfe P., Crowder H.* Validation of subgradient optimization. Math. Program., 1974, **6**, № 1, p. 62—88.
492. *Hestenes M. R.* The conjugate gradient method for solving linear system. In: Proceeding of Symposia on Applied Mathematics, v. VI, Numerical Analysis, New York, 1956, p. 83—102.
493. *Hestenes M. R.* Multiplier and gradient methods. J. of Opt. Theory and Appl., 1969, **4**, № 5, p. 303—320.
494. *Hestenes M. R., Stiefel E.* Methods of conjugate gradients for solving linear systems. J. Res. Nation. Bureau Stand., 1952, **49**, № 6, p. 409—436.
495. *Hohenbalken Von Balder.* Simplicial decomposition in nonlinear programming algorithms. Math. Program., 1977, **13**, № 1, p. 49—63.
496. *Huard P.* Rezolution of mathematical programming with nonlinear contraints by the method of centres. In: Nonlinear Programming / Ed. J. Abadie. Amsterdam : North Holland Publ. Co., 1967, p. 208—219.
497. *Huard P.* Programmation matématique convexe. Rev. Française Informat. Rech. Operationnele, 1968, № 7, p. 43—59.
498. *Huang H. Y., Levy A. V.* Numerical experiments on quadratically convergent algorithm for function minimization. J. of Optim. Theory and Appl., 1970, **6**, № 3, p. 269—282.
499. *Jocobsen St. E., Torabi M.* A global minimization algorithm for a class of one dimensional functionals. J. Math. Anal. and Appl., 1978, **62**, № 2 , p. 310—324.
500. *Kalfond P., Ribiere G., Sogno J. C.* A method of feasible directions using projection operators. In: Proc. IFIP Congr. 68, Edinburgh, August, 1968.
501. *Keefer J.* Sequentional minimax search for a maximum. Proc. Amer. Math. Soc., 1953, **4**, № 3, p. 502—506.
502. *Keefer J., Wolfowitz J.* Stochastic estimation of maximum of a regression function. Ann. of Math. Stat., 1952, **23**, p. 463—466.
503. *Kelley J. E.* The cutting plane method for solving convex programs. SIAM Journal, 1958, **6**, № 1, p. 15—22.
504. *Kelley J. E.* The cutting plane method for solving convex programs. SIAM Journal, 1960, **8**, № 4, p. 703—712.
505. *Klessig R.* A general theory of convergence for constrained optimization algorithms that use antizigzagging provisions. SIAM, J. Contr., 1974, **12**, № 4, p. 598—608.
506. *Klessig R., Polak E.* Effecient implementations of Polak Ribiere conjugate gradient algorithm. In: Electronics Research Laboratory Memo № M279. Berkeley, University of California, Aug. 1970.
507. *Konno Hiroshi* Maximization of a convex quadratic function under linear constraints. Math. Program., 1976, **11**, № 2, p. 117—127.
508. *Kowalik J., Osborne M. R.* Methods for unconstrained optimization problems : American Elsevier. New York, 1968, p. 120.
509. *Kowalik J., Osborne M. R., Ryan D. M.* A new method for constrained optimization problems. Oper. Res., 1969, **17**, № 6, p. 973—983.
510. *Kuhn H. W., Tucker A. W.* Nonlinear programming. In: Proceeding of the Second Berkeley Symposium on Mathematical Statistics and Probability / Ed. J. Neyman. Uniwersity of California Press, 1951, p. 481—493.
511. *Laenberger D.* Convergence rate of penalty function scheme. J. of Opt. Theory and Appl., 1971, **7**, № 1, p. 39—51.
512. *Lemarechal C.* An extention of Davidon methods to nondifferentiable problems. Math. Programming Study, 1975, **3**, p. 95—105.

513. *Litvakov B. M.* On a class of Robbins Monro procedure. Inform. Sci., 1973, **6**, № 1, p. 33—48.
514. *Lootsma F. A.* Hessian matrices of penalty functions for solving constrained optimization problems. Philips Res. Rept., 1969, **24**, p. 322—331.
515. *Marks B. R., Wright G. P.* A general inner approximation algorithm for non-convex mathematical programs. Oper. Res., 1978, **26**, № 4, p. 681—683.
516. *Mathematical* programming. Study 3. Nondifferentiable optimization / Ed. M. L. Balinski, P. Wolfe. Amsterdam : North Holland Publ. Co., 1975.— 178 p.
517. *Mayne D. Q., Polak E.* Feasible directions algorithms for optimization problems with equality and inequality constraints. Math. program., 1976, **11**, № 1, p. 67—80.
518. *McCormick G. P.* A second order method for the linearly constrained nonlinear programming problem. In: Nonlinear Programming / Eds. J. B. Rosen, O. L. Mangasarian, K. Ritter. London; New York : Academic Press, 1972, p. 209—243.
519. *McCormick G. P.* Attemps to calculate global solutions of problems that may have local minima. In: Numerical Methods for Nonlinear Ortimization / Ed. F. A. Lootsma. London; New York : Academic Press, 1972, p. 209—221.
520. *Meyers G. E.* Properties of the Conjugate Gradient and Davidon Methods. J. of Optim. Theory and Appl., 1968, **2**, p. 209—219.
521. *Meyer Gegard G. L.* Accelerated Frank Wolfe algorithms. SIAM J. on Contr., 1974, **12**, № 4, p. 655—663.
522. *Mifflin R.* On the convergence of the logarithmic barrier function. In: Numerical Methods for Nonlinear Optimization / Ed. F. A. Lootsma. London; New York : Academic Press, 1972.
523. *Mifflin R.* An algorithm for constrained optimization with semismooth functions. Math. Operations Res., 1977, № 2, p. 191—207.
524. *Mifflin R.* A stable method for solving certain constrained least squares problems. Math. Program., 1979, **16**, № 2, p. 141—158.
525. *Mine H., Fukushima M.* Penalty function theory for General convex programming problems. J. of Optim. Theory and Appl., 1978, **24**, № 2, p. 287—301.
526. *Mine H., Olina K., Fukushima M. A.* «Conjugate» interior penalty method for certain convex programs. SIAM J. Contr. and Opt., 1977, **15**, № 5, p. 747—755.
527. *Murphy F.* A class of exponential Penalty functions. SIAM J. on Contr., 1974, **12**, p. 679—687.
528. *Murray W.* Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions. J. of Optim. Theory and Appl., 1971, **7**, p. 189—196.
529. *Mutsumato T.* On a class of min-max functions. Trans. Soc. Instrums. Contr. Eng., 1975, **11**, № 3, p. 322—327.
530. *O'Neill Richard P., Widhelm William B.* Acceleration of Lagrangian column generation algorithms by penalty function methods. Management Sci., 1976, **23**, № 1, p. 50—58.
531. *Von Neumann J.* A numerical method to determine optimum strategy. Nav. Res. Logist. Quart., 1954, **1**, № 2, p. 109—115.
532. *Nguyen V. H., Strodiot J. J.* On the convergence rate for a penalty function method of exponential type. J. of Optim. Theory and Appl., 1979, **27**, № 4, p. 495—508.
533. *Parikh S. C.* Approximate cutting planes in nonlinear programming. Math. program., 1976, **11**, № 2, p. 194—198.
534. *Pearson J. D.* Variable metric method of minimization. Comput. J., 1969, **12**, № 2, p. 171—178.
535. *Pietszykowski T.* An iteration method of linear programming. Warczawa : Zaklad Aparatow Mathematycznich PAN, Prace A, 1960, № 5.
536. *Polak E.* On the convergence of optimization algorithms. Rev. Francaise Informat. Rech. Operationelle, 16—R1, 1969, p. 17—34.
537. *Polak E.* A modified secant method for unconstrained minimization. Math. Program. 1974, **6**, p. 264—280.

538. *Polak E.* On the global stabilization of locally convergent algorithms for optimization and root finding. In: Proc. 6-th World Congress IFAC; Booton / Cambridge, Massachusetts, USA, 1977, p. 1—5.
539. *Polak E., Ribière G.* Note sur la convergence de methodes de directions conjugées. Rev. Française Inform. Rech. Operationelle, 1969, 16—R1, p. 35—43.
540. *Powell M. J. D.* An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput. J., 1964, **7**, № 2, p. 155—162.
541. *Powell M. J. D.* A method for nonlinear constraints in minimization problems. In: Optimization, Chapter 19. London; New York : Acad. Press, 1969, p. 283—298.
542. *Powell M. J. D.* Some convergence properties of the conjugate gradient method. Math. program., 1976, **11**, № 1, p. 42—49.
543. *Powell M. J. D.* A survey of numerical methods for unconstrained optimization. SIAM Rev., 1970, **12**, № 1, p. 79—97.
544. *Powell M. J. D.* Algorithms for nonlinear constraints that use langrangian functions. Math. Program., 1978, **14**, № 2, p. 224—248.
545. *Rao S. R., Mitra S. K.* Generalized inverse of matrices and its applications. New York, John Wiley, 1972.
546. *Reeves Gary R.* Global minimization in nonconvex all quadratic programming. Management Sci., 1975, **22**, № 1, p. 76—86.
547. *Robbins H., Monro S.* A stochastic approximation method. Ann. Math. Stat., 1951, **22**, № 3, p. 400—407.
548. *Robinson S. M.* A quadratically convergent algorithm for general nonlinear programming problems. Math. Program., 1972, **3**, № 2, p. 145—156.
549. *Rockafellar R. T.* The Multiplier Method of Hestenes and Powell applied to convex programming. J. of Optim. Theory and Appl., 1973, **12**, № 6, p. 6.
550. *Rockafellar R. T.* Augmented lagrang multiplier functions and dual nonconvex programming. SIAM J. Contr., 1974, **12**, № 2, p. 268—285.
551. *Rosen J. B.* The gradient projection method for nonlinear programming, Part I, Linear Constraints. SIAM J. Appl. Math., 1960, **8**, № 1, p. 181—217.
552. *Rosen J. B.* The gradient projection method for nonlinear programming, Part II, Nonlinear Constraints. SIAM, J. Appl. Math., 1961, **9**, p. 514—532.
553. *Rozenbrock H. H.* An automatic method for finding the greatest or least value of the function. Comput. J., 1960, **3**, p. 175—184.
554. *Saaty T. L.* Coefficient perturbation of a constrained extremum. Oper. res., 1959, **7**, № 3.
555. *Sasal Hitoshi.* An interior penalty method for minimax problems with constraints. SIAM J. on Control., 1974, **12**, № 4, p. 643—649.
556. *Schubert L. K.* Modification of quasi Newton method for nonlinear equations with a sparse Jacobian. Math. Comput., 1970, **24**, № 1, p. 27—30.
557. *Sharp J. Franklin, Wagner Michael H.* Generalized programming by linear approximation of the dual gradient (equality constraints). Nav. Res. Logist. Quart., 1975, **22**, № 1, p. 145—154.
558. *Shultz Hilbert K.* A Kuhn Tucker algorithm. SIAM J. Contr., 1973, **11**, № 3, p. 438—445.
559. *Singh C.* Sufficient optimality criteria in nonlinear programming for generalized equality-inequality constraints. J. of Optim. Theory Appl., 1977, **22**, № 4, p. 631—635.
560. *Stong R. E.* A note on the sequential unconstrained minimization technique for non linear programming Management Sci., 1965, **12** (1), p. 142—144.
561. *Strodiot J. J., Nguyen V. H.* An exponential penalty method for non differentiable minimax problems with general constraints. J. of Optim. Theory and Appl., 1979, **27**, № 2, p. 205—219.
562. *Ting A. C.* On the convergence of a class derivative free minimization algorithms. J. of Optim. Theory and Appl., 1977, **22**, № 4, p. 487—508.
563. *Topkis D. M., Veinott A. F.* On the convergence of some feasable directions algorithms for nonlinear programming. SIAM J. on Contr., 1967, **5**, № 2, p. 268—279.

564. *Tremolières R.* Méthode des centres a troncature variable. Elec. FR. Bull. Dir. Etudes Rech. Ser. C-Math. Inform. 1968, **2**, p. 57—64.
565 *Truemper K.* Note on finite convergence of exterior penalty functions. Manag. Sci., 1975, **21**, № 5, p. 600—606.
566. *Wolfe Ph.* The simplex method for quadratic programming. Econometrica, 1959, **28**, № 3, p. 382—398.
567. *Wolfe Ph.* Accelerating the cutting plane method for non linear programming. SIAM Journal, 1961, **9**, p. 481—488.
568. *Wolfe Ph.* A method of conjugate subgradients for minimizing nondifferentiable functions. In: Mathematical Programming. Study **3**. Non differentiable optimization / Ed. M. L. Balinski, P. Wolfe. Amsterdam : North-Holland Publ. Co., 1975, p. 145—173.
569. *Wolfe Ph.* Convergence theory in nonlinear programming. In: Integer and nonlinear programming / Ed. J. Abadie. Amsterdam : North Holland Publ. Co., 1970, p. 1—36.
570. *Zangwill W. I.* Minimizing a function without calculating derivatives. Comput. J., 1967, **10**, № 3, p. 293—296.
571. *Zangwill W. I.* Nonlinear programming via penalty functions. Management Sci., 1967, **13**, № 5, p. 344—358.
572. *Zangwill W. I.* Convergence conditions for nonlinear programming. Management Sci., 1969, **16**, № 1, p. 1—13.
573. *Zoutendijk G.* Nonlinear programming (a numerical survey). SIAM J. on Contr., 1966, **4**, № 1, p. 194—210.
574. *Zoutendijk G.* Nonlinear programming, computational methods. In: Integer and Nonlinear Programming / Ed. J. Abadie. Amsterdam : North Holland Publishing Co., 1970, p. 120.
575. *Zwart Philip B.* Global maximization of a convex function with linear inequality constraints. Oper. Res., 1974, **22**, № 3, p. 602—609.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

Адаптивный метод 21, 35
y_0-адекватная математическая модель 14
$B\delta$-адекватная модель 43
ε-активное множество индексов 299
Алгоритм
вычисления ближайшей к началу координат точки многогранника 171
Кифера — Вольфовица 71
— — нормализованный 72
определения принадлежности начала координат многограннику 170
оптимальный 23, 49, 52
переменной метрики 37, 98
Франка — Вулфа 376
— — ускоренный 377

Базис опорного решения 193, 204, 229, 235
— — — почти допустимого 204
Базисная матрица 194
Базисные компоненты опорного решения 194

Векторы
искусственные 197
сопряженные 38, 99
Внешние штрафные функции 279, 281
Внешних штрафных функций метод 281, 283, 287
— — — — модифицированный 284
Внутренней аппроксимации метод 387
Внутренние штрафные функции 279, 289
Внутренних штрафных функций метод 281, 289, 290, 292
Внутренняя точка 392
Возможное направление 297
Возможных направлений метод 33, 41, 297, 301, 307, 309, 313
— — — модифицированный 307
— — — с квадратичным поиском 305
— — — стохастический аналог 311
Второго порядка метод 446
Выпуклого программирования задача 15, 346

Геометрической прогрессии (линейная) скорость сходимости метода (алгоритма) 33
Гибридный метод проекции градиента 40, 269, 270, 277
— — стохастический 414
Глобально сходящийся метод 403
Глобального поиска метод 63
— — — рандомизированный 65
Градиент функции 27
— — обобщенный (субградиент) 29
Градиентный метод 33, 73, 76—81, 464
— — обобщенный 187, 467
— — с растяжением пространства 146, 149
Граничная точка 392

Данцига — Вулфа метод разложения 219
Двойственная (сопряженная) задача 203, 212, 228, 248, 250, 398
Двойственный метод 353, 398
— — переменной метрики 401
— — приближенный 398
Двойственный симплекс-метод 203
Двойственных направлений метод 87
— — — модифицированный 89
Двусторонних оценок метод 214
Допустимых направлений конус 28
Достаточное условие оптимальности 29, 30
Дроби Фибоначчи 96

Задача
многоцелевого управления 19
оптимизации 13, 14
— безусловная 14, 73, 136
—двойственная (сопряженная) 203, 228, 248, 250, 398, 212
— дискретная 15
— минимаксная 17
— — дискретная 167, 438, 440, 444
— — непрерывная 178, 180, 449
— — стохастическая 188, 189, 461
— многоэкстремальная 15
— многоэтапная 18
— при неполных данных 14

— прямая 203, 228, 248, 250, 398
— с бесконечным числом ограничений 424
— условная 14, 192, 269, 438
отыскания седловых точек 187, 464, 468
поиска максимина 450, 452, 457, 458
предельная экстремальная 339, 373
программирования блочного 219, 222 223, 227
— выпуклого 15, 346
— квадратичного 427, 428, 431, 432, 435
— линейного 15, 192
— — каноническая 192
— — — вырожденная 194
— — — невырожденная 194, 204, 230
— — с двусторонними ограничениями 229
— нелинейного 269, 438
— параметрического 16, 253, 257
— стохастического 16, 412, 420, 421, 461
Золотого сечения метод 48

Искусственные
векторы 197
переменные 197
Итеративный метод 240, 242, 243, 247, 250

Каноническая задача линейного программирования 192
— — — — вырожденная 194
— — — — невырожденная 194
Касательных метод 56
Квадратичная скорость сходимости метода (алгоритма) 33
— функция 97, 104
Квадратичного программирования задача 427, 428, 431, 432, 435
Квадратичной аппроксимации метод 58, 353
Квазивыпуклая функция 24
Квазиградиент функции 157
ε-квазиградиент функции 160
Квазиградиентный метод 41, 121—126, 156—158, 405
— — стохастический 188—190
ε-квазиградиентный метод 159, 160
Квазиньютоновский метод 37, 325
(ε, μ)-квазистационарная точка 443
Квазирешение разрывных функций 165
Квазифейеровская последовательность случайная 408
Кифера — Вольфовица алгоритм 71
Классические условия шаговых множителей 137
Комбинированный мето
стохастических градиентов и штрафных функций 416
штрафных функций 293
Конечно-разностный метод 165
Конус допустимых направлений 28
Коэффициент штрафа (штрафной коэффициент) 278
Куна — Таккера
теорема 29, 30
точка 30
Кусочно-кубической аппроксимации метод 61

Лагранжа
множители 30
функция 30, 350, 352
— модифицированная 356, 365, 369, 398
Линеаризации метод 41, 327, 330, 332, 335, 336
— — в предельных экстремальных задачах 339
— — практически реализуемый на ЭВМ 333
— — стохастический 337
Линейная (геометрической прогрессии) скорость сходимости метода (алгоритма) 33
Линейного программирования задача 15, 192
— — — каноническая 192
— — — — вырожденная 194
— — — — невырожденная 194
— — — с двусторонними ограничениями 229
Липшица
константа 25
условие 25, 75, 84
функция 25
Локального случайного поиска метод 151, 152, 153
Локальное решение 22
Локальной оптимизации метод 22
Локальных вариаций метод 128

Максимин 450, 452
Математическая модель 14
— — y_0-адекватная 14
— — реализуемая 14
Матрица базисная 194
Метод
адаптивный 21, 35
аппроксимации 22
— внутренней 387
— квадратичной 58, 353
— кусочно-кубической 61
возможных направлений 33, 41, 297 301, 307, 309, 313
— — модифицированный 307
— — с квадратичным поиском 305
— — стохастический аналог 311

второго порядка 446
гибридный
— проекции градиента 274
— стохастический 414
глобально сходящийся 403
глобального поиска 63
— — рандомизированный 65
градиентный 33, 73, 76—81, 464
— обобщенный 187, 467
— с растяжением пространства 146, 149
двойственный 353, 398
— переменной метрики 401
— приближенный 398
двойственных направлений 87
— — модифицированный 89
двусторонних оценок 214
декомпозиции развязывающей 41
золотого сечения 48
использующий аналог функции Ляпунова 116
— стохастические автоматы 68
— функцию Лагранжа 350, 351, 353, 354
— — — модифицированную 355, 357
итеративный 240, 242, 243, 247, 250
касательных 56
квазиградиентный 41, 121—126, 156—158, 405
— стохастический 188—190
ε-квазиградиентный 159, 160
квазиньютоновский 37, 325
комбинированный, стохастических градиентов и штрафных функций 416
конечно-разностный 165
линеаризации 41, 327, 330, 332, 335, 336
— в предельных экстремальных задачах 339
— практически реализуемый на ЭВМ 333
— стохастический 337
локальной оптимизации 22
локальных вариаций 128
минорант 25
многошаговый 143, 371
Монте-Карло 21
нагруженного функционала 360, 361
— — приближенная схема 363
наискорейшего спуска 73
— — модифицированный 74
невязок 458
Ньютона 55, 322, 354
— при наличии возмущений 323
— с регулировкой шага 322
Ньютона — Канторовича 37, 82
— модифицированный 84, 85
— — обобщенный 82
обобщенного градиентного спуска 136—142, 187
— — — многошаговый 143, 372
обобщенный беспараметрический внешней точки 444
— — — — ускоренный вариант 445
обобщенных почти градиентов 161
опорный 261
— мультипликативный 268
— прямой 262
— обратной матрицы 265
оптимальный поиска экстремума выпуклых функций 52
— — — унимодальных функций Липшица 49
отсечений 21, 24, 343
— с растяжением пространства 346
перебора полного 21
— простого 72
поиска интервала наибольших значений многоэкстремальных функций 66
покоординатного спуска детерминированный 40, 388
— — случайный 390
последовательного сокращения невязок 211
— уточнения оценок 204
последовательных приближений 22, 32, 40, 169—177, 438—442
— — сеточный 178, 449
проектирования обобщенного градиента 371, 373
— — — многошаговый 372
— стохастических квазиградиентов 407
проекции градиента 40, 269, 270
— — гибридный 274
— — при наличии возмущений 277
прямой 421
псевдоградиентный 154
псевдообратных операторов 106
развязывающей декомпозиции 41
разложения Данцига — Вулфа 219
релаксационный дискретный 393
— непрерывный 392
растяжения пространства 37, 146, 149
самонастраивающихся программ 129
секущей 55
случайного поиска 133, 423
— — локального 151, 152, 153
сокращения невязок стохастический 410, 412
сопряженных градиентов 92, 94, 97, 378, 427, 428, 431
— — модифицированный 432
— — с восстановлением 93
— — стохастический аналог 380
— направлений 99, 100, 102, 105

— — модифицированный 105, 432
стохастических квазиградиентов 121, 188, 190, 405, 452, 461
— — модифицированный 462
управляющих последовательностей 381, 383
условного градиента реализуемый 375
усреднения направлений спуска 164, 417, 420
фейеровских приближений 395, 396
Фибоначчи 46
— модифицированный 47
центров 316
— модифицированный 316
— — реализация 318, 319
— чебышевских 320
— — модифицированный 321
штрафных оценок 297, 301, 365, 367, 368
— функций внешних 281, 283, 287
— — — модифицированный 284
— — внутренних 281, 289, 290, 292
— — комбинированный 293
— — общий 278
штрафов 39, 450
— стохастический 295
— экспоненциальный 357, 468
экстремального базиса 182, 184, 185
эллипсоидов 25
Эрроу — Гурвица детерминированный 180
— — стохастический 181
Минимаксная задача 17
— — дискретная 167, 438, 440
— — непрерывная 178, 449
— — стохастическая 188, 189, 461
Минимизирующая последовательность 32
Минорант метод 25
Многоцелевого управления задача 19
Множество
выпуклое 28
— сильно 28, 376
— строго 28
индексов ε-активное 299
квазиградиентов 31
оптимальности базиса 253, 257
седловых точек 187, 464
— — условие устойчивости 464, 467
субградиентов 29
Множители
Лагранжа 30
нормирующие 122, 371, 287
шаговые 33, 74, 76, 77, 80, 81, 83, 95, 96, 122, 124, 287
— классические условия 137

Нагруженного функционала метод 360, 361
— — — приближенная схема 363
Нагруженный функционал 360, 362
Наискорейшего спуска метод 73
— — — модифицированный 74
Направление
наибыстрейшего убывания 27
ε-наискорейшего спуска 169
Небазисные компоненты опорного решения 194
Невырожденная задача линейного программирования 194, 204, 230

Невырожденный опорный план 262
Невязок метод 458
Необходимое условие оптимальности 29, 30, 31, 168, 283
Ньютона метод 55, 322, 354
— — при наличии возмущений 323
— — с регулировкой шага 322
Ньютона — Канторовича метод 37, 82
— — — модифицированный 84, 85
— — — обобщенный 82

Обобщенного градиентного спуска метод 136—144, 187
— — — — многошаговый 143, 372
Обобщенных почти градиентов метод 161
Обобщенный
градиент (субградиент) функции 29
беспараметрический внешней точки метод 444
— — — — ускоренный вариант 445
симплекс-метод 216
Обратной матрицы опорный метод 265
Общий метод штрафных функций 278
Ограничения задачи
двусторонние 229
кусочно-линейные 396
линейные 192, 270
общие 192
простой структуры 438
прямые 192
типа неравенств 281, 311, 313, 320
— равенств 283
— смешанного 192, 301, 307, 332, 392
Оператор
развязывающий 42
растяжения пространства 146
релаксации 383
сжатия пространства 146
Опора задачи 262
Опорное решение 193, 204, 229, 235
— — вырожденное 194, 204
— — невырожденное 194, 204, 230, 235
— — почти допустимое 203
Опорный
метод 261
— мультипликативный 268
— прямой 262
— обратной матрицы 265

план 262
Оптимальное решение 13
Оптимальный
алгоритм 23, 49, 52
метод поиска экстремума функций выпуклых 52
— — — — унимодальных Липшица 49
Оптимизации задача 13, 14
Отображение фейеровское 395
Отсечений метод 21, 24, 343
— — с растяжением пространства 346

Параметрического программирования задача 16, 253, 257
Перебора метод
полного 21
простого 72
Переменной метрики алгоритм 37, 98
Переменные искусственные 197
Покоординатного спуска метод 40
— — — детерминированный 388
— — — случайный 390
Последовательного сокращения невязок метод 211
— уточнения оценок метод 204
Последовательность
квазифейеровская случайная 408
сеток плотная на множестве 179, 462
управляющая 382
— согласованная с оператором 385
Последовательных приближений метод 22, 32, 40, 169—177, 438—442
— — — сеточный 178, 449
Правило множителей Лагранжа 366
Прерывания функция 319
Проектирования обобщенного градиента метод 371, 373
— — — — многошаговый 372
— стохастических квазиградиентов метод 407
Проекции градиента метод 40, 269, 270
— — — гибридный 274
— — — при наличии возмущений 277
Производная функции 27
— — по направлению 27
Прямой метод 421
Псевдоградиент функции 154
Псевдоградиентный метод 154
Псевдообратных операторов метод 106
Псевдообращение матриц 107

Равномерно выпуклая (вогнутая) функция 471
Развязывающей декомпозиции метод 41
Развязывающий оператор 42
Разложения Данцига — Вулфа метод 219
Рандомизированный алгоритм глобального поиска 65
Растяжения пространства
метод 37, 146, 346
оператор 146
Реализуемая математическая модель 14
Регулярная точка 30
Регулярности ограничений условие 393
Релаксации оператор 383
Релаксационный метод
— — дискретный 393
— — непрерывный 392
Решение
локальное 22
стационарное (стационарная точка) 32, 170, 440, 449, 455
опорное 193, 204, 229, 235
оптимальное 13

Самонастраивающихся программ метод 129
Сверхлинейная скорость сходимости метода (алгоритма) 33
Седловая точка 30
Седловое множество 187, 464
— — условие устойчивости 464, 467
Секущей метод 55
Сжатия пространства оператор 146
Сильно выпуклая функция 28
Сильно выпуклое множество 28, 376
Симплекс-метод 193, 199
— — двойственный 203
— — модифицированный 201, 229, 235
— — мультипликативный 203
— — обобщенный 216
Скорость сходимости метода (алгоритма)
геометрической прогрессии, или линейная 33
квадратичная 33
k-го порядка 33
сверхлинейная 33
Слабо выпуклая (вниз) функция 156
Слейтера условие 30, 350
Случайного локального поиска метод 151—153
Случайного поиска метод 133, 423
Сокращения невязок стохастический метод 410, 412
Сопряженные векторы 38, 99
Сопряженных градиентов метод 92, 94, 97, 378, 427, 428, 431
— — — модифицированный 432
— — — с восстановлением 93
— — — стохастический аналог 380
Сопряженных направлений метод 99, 100, 102, 105
— — — модифицированный 105, 432
Средняя квадратическая сходимость метода (алгоритма) 120

Стационарная точка (стационарное решение) 170, 440, 449, 455, 32
ε-стационарная точка 174
Стохастический квазиградиент 121, 405
Стохастических квазиградиентов метод 121, 188, 190, 405, 452, 461
— — — модифицированный 462
Стохастического программирования задача 412, 420, 421, 461
Строго выпуклая функция 28
— выпуклое множество 28
ε-субградиент функции 145
Субградиент функции (обобщенный градиент) 29
Сходимость метода (алгоритма)
— — в среднем 117
— — почти наверное (с вероятностью единица) 120, 124, 411, 424
— — по вероятности 126, 128, 368
— — средняя квадратическая 120

Теорема Куна — Таккера 29, 30
Точка
внутренняя 392
граничная 392
(ε, μ)-квазистационарная 443
Куна — Таккера 30
локального минимума 30
регулярная 30
стационарная (стационарное решение) 32, 170, 440, 449, 455

Унимодальная функция 49
Управляющая последовательность 382
— — согласованная с оператором 385
Управляющих последовательностей метод 381, 383
Условие
Липшица 25, 75, 84
оптимальности достаточное 28, 29, 30
— необходимое 27—31
— поведения автомата в случайной стационарной среде 70
регулярности ограничений 393
Слейтера 30, 350
строгой дополняющей нежесткости 326, 359
Условия классические шаговых множителей 137
Условного градиента реализуемый метод 375
Усреднения направлений спуска метод 164, 417, 420

Фейеровских приближений метод 395, 396
Фейеровское отображение 395
Фибоначчи
дроби 96
метод 46
— модифицированный 47
числа 49
Франка — Вулфа алгоритм 376
— — — ускоренный 377
Функционал нагруженный 360, 362
Функция
вогнутая равномерно 471
выпуклая 28
— равномерно 471
— сильно 28
— строго 28
дифференцируемая 27
квадратичная 97, 104
квазивыпуклая 24
квазидифференцируемая 31, 32
Лагранжа 30, 350, 352
— модифицированная 356, 365, 369, 398
липшицева 25
максимизируемая 15
минимизируемая 15
прерывания 319
унимодальная 49
целевая 15
штрафа (штрафная) 39, 278
D-функция 177
Q-функция 292

Целевая функция 15
Центров метод 316
— — модифицированный 316
— — — реализация 318, 319
— — чебышевских 320
— — — модифицированный 321

Числа Фибоначчи 49

Шаговые множители 33, 74, 76, 77, 80, 81, 83, 95, 96, 122, 124, 287
Штрафа коэффициент (штрафной коэффициент) 278
Штрафная функция (функция штрафа) 39, 278
Штрафных оценок метод 297, 301, 365, 367, 368
Штрафных функций метод
— — — внешних 281, 283, 287
— — — — модифицированный 284
— — — внутренних 281, 289, 290, 292
— — — комбинированный 293
— — — общий 278
Штрафов метод 39, 450
— — стохастический 295
— — экспоненциальный 357, 468

Экспоненциальных штрафов метод 357, 468
Экстремального базиса метод 182, 184, 185
Эллипсоидов метод 25
Эрроу — Гурвица метод
— — — детерминированный 180
— — — стохастический 181

Иван Васильевич Бейко
Борис Николаевич Бублик
Петр Николаевич Зинько

Методы и алгоритмы решения задач оптимизации

Редакторы *В. И. Денисюк, Л. Г. Любимова, М. С. Хойнацкий*
Переплет художника *В. А. Гурлева*
Художественный редактор *С. П. Духленко*
Технический редактор *Л. Ф. Волкова*
Корректор *Н. В. Волкова*

Информ. бланк № 7536

Сдано в набор 30.12.81. Подп. в печ. 27.07.83. БФ 03179. Формат 60×90/16. Бумага книж.-журн. Лит. гарн. Выс. печать. 32 печ л. 32 кр.-отт. 31,75 уч.-изд. л. Тираж 5000 экз. Изд. № 5280. Зак. № 3-341. Цена 2 р.

Головное издательство издательского объединения «Вища школа», 252054, Киев-54, ул. Гоголевская, 7.

Отпечатано с матриц Головного предприятия РПО «Полиграфкнига» на книжной фабрике «Коммунист», 310012, Харьков-12, Энгельса, 11.